AFRL-IF-RS-TR-2005-382
Final Technical Report
November 2005

ULTRASCALABLE TECHNIQUES APPLIED TO THE GLOBAL INTELLIGENCE COMMUNITY INFORMATION AWARENESS COMMON OPERATING PICTURE (IA COP)

SRI International

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-382 has been reviewed and is approved for publication

APPROVED: /s/

WLADIMIR TIRENIN
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** NOVEMBER 2005	**3. REPORT TYPE AND DATES COVERED** Final Sep 03 – Jul 05

4. TITLE AND SUBTITLE
ULTRASCALABLE TECHNIQUES APPLIED TO THE GLOBAL INTELLIGENCE COMMUNITY INFORMATION AWARENESS COMMON OPERATING PICTURE (IA COP)

5. FUNDING NUMBERS
C - F30602-03-C-0234
PE - 31011G
PR - B104
TA - 00
WU - 01

6. AUTHOR(S)
Alfonso Valdes and
Jim Kadte

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SRI International
333 Ravenswood Avenue
Menlo Park California 94025

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFGB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-382

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Wladimir Tirenin/IFGB/(315) 330-1871/ Wladimir.Tirenin@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

The focus of this research is to develop detection, correlation, and representation approaches to address the needs of the Intelligence Community Information Awareness Common Operating Picture (IA COP). The approaches build on existing enterprise information security tools where appropriate, and depart from these traditional methods where required. In particular, the requirement to scale to large networks and data repositories is the primary driver for technical innovation.

We explored the following areas:

- Representation of network observables to enable signature-free detection at various network scales. Mining these observables to detect emerging phenomena, departures from trends, and anomalies visible at multiple sites.
- A departure from the current incident-centric approach to intrusion alert correlation toward an entity centric "dossier" methodology.
- Incorporation of techniques from nonlinear dynamical systems to identify, for example, loci of unusual activity.

14. SUBJECT TERMS
Situation Awareness, Large Scale Network Event Correlation, Visualization, Netflow Analysis

15. NUMBER OF PAGES
50

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1 Purpose of Report.. 1

2 Summary of Technical Objectives and Approach 1

2.1 Ultrascalable Techniques for Pattern and Trend Analysis 1

2.2 Dossier Correlation.. 2

2.2.1 N-Gram Port Analysis 3
2.2.2 Experiments on Blackhole Traffic Patterns.. 4
2.2.3 N-Gram IDS Alert Production Analysis.. 7
2.2.4 Experiments with IDS Alerts... 7

2.3 Nonlinear Dynamical Analysis... 8

3 Technology Transition 13

Appendix A. Entropy Characterization of Propagating Internet Phenomena................. 15
Appendix B. Data Cube Indexing of Large-scale Infosec Repositories 27
Appendix C. Improved Epidemic Path Predictability in Complex Networks 37

List of Figures

Figure 1: Sample Destination Port N-Gram Cound... 5
Figure 2: N-Gram Port Combinatorial Graph.. 6
Figure 3: Correlation Versus Node Degree 10
Figure 4: Infection Fraction as a Function of Node Degree, Density, and Distance to Seed 11
Figure 5: Evolution Infected Fraction Versus Node Degree (top) and RWC (bottom)... 12
Figure 6: Screenshot of Scalable Visualization, CIAC Network.................................... 14

List of Tables

Table 1 - N-Grams of Observed Alert IDs.. 8

1 Purpose of Report

This report is submitted as CDRL A009 under Contract F30602-03-C-0234, Ultrascalable Techniques Applied to the Global Intelligence Community Information Awareness Common Operating Picture (IA COP).

Views presented here are those of the authors and do not necessarily reflect the views of the sponsoring organization.

2 Summary of Technical Objectives and Approach

The focus of this research is to develop detection, correlation, and representation approaches to address the needs of the Intelligence Community Information Awareness Common Operating Picture (IA COP). The approaches build on existing enterprise information security tools where appropriate, and depart from these traditional methods where required. In particular, the requirement to scale to large networks and data repositories is the primary driver for technical innovation.

We explored the following areas:

- Representation of network observables to enable signature-free detection at various network scales. Mining these observables to detect emerging phenomena, departures from trends, and anomalies visible at multiple sites.
- A departure from the current incident-centric approach to intrusion alert correlation toward an entity centric “dossier” methodology.
- Incorporation of techniques from nonlinear dynamical systems to identify, for example, loci of unusual activity.

This report is organized as follows. In the remainder of this section, we provide a high-level summary of the objectives and findings of investigations in each of the above topic areas. We then describe our efforts toward technology transition with potential partners in the Department of Energy (DOE) and the National Security Agency (NSA). This is followed by a series of technical appendices, adapted from papers we have submitted to workshops, conferences, and journals, providing a more in-depth technical development of selected topics.

2.1 Ultrascalable Techniques for Pattern and Trend Analysis

This topic area explored algorithms for detection and representation in large-scale networks, considering in particular signature-free techniques to address emergent phenomena. We examined the following sources of data:

- Packets rejected at the firewall between our departmental network and the Internet. These packets were often the result of scanning worms.
- Of packets rejected at the firewall, specifically focusing on packets to unused address space. This is analogous to the “black hole” and “Internet telescope” analysis performed by other investigators.

- The Hackfest data set, generated in support of the ARDA P2INGS program.
- Peering point data in NetFlow format.
- Patterns of alert signatures from IDS alert repositories (in this case, the EMERALD NIDS appliance monitoring our departmental network.

The black hole data and the alert repository analysis provided a complementary approach to the analysis performed under dossier correlation, discussed further below.

Among the principal findings in this topic area, we observed that the entropy of the source address and destination port distributions, in particular, exhibit dramatic and characteristic change in the early stages of a large-scale propagating event. Specifically, the range of addresses making connections to a targeted host or network becomes significantly more random, while the distribution of ports accessed becomes less random. This results in a detectable increase in the entropy of source IP addresses and decrease in the entropy of destination ports.

We implemented a detector based on this observation, featuring an innovative and computationally efficient way to continuously track entropy of key packet header parameters. This technique was more fully developed from earlier work sponsored by DARPA. By efficient approximations of otherwise expensive log functions as well as parsimonious pruning of the state space, this detection component has comparatively modest processor and memory requirements, and can easily keep up with traffic at our laboratory gateway firewall. We conjecture that it can keep up with traffic at gateways to much larger networks, but have not had the opportunity to prove this in a live setting.

Additionally, we developed a technique whereby key variables (source and destination address, source and destination port, or IDS alert identifier) are hashed to a finite range, and intensity is shown on two-dimensional displays. Propagating events have a characteristic appearance in displays of traffic, or rejected traffic, at a source/destination pair or from all sources to a particular set of ports. Patterns of alerts also characterized scanning activity of a more subtle nature than loud propagating worm events.

These results were presented as abstracts and workshop papers, with the most comprehensive summary of each result included as appendices to the present report.

2.2 Dossier Correlation

A second area of study within this research project has involved the development of techniques to summarize the common behavioral characteristics of malicious remote intruders, with the intent of associating intruders under a common behavior profile, or *dossier*. In *dossier-based* correlation, a correlation digest is constructed per source or target to capture the history and severity of behavior exhibited by each entity. For example, in the context of large-scale alert repository analysis, sources may be grouped under a common dossier representing the sequence of alerts that they have triggered. In the analysis of firewall logs or egress router packet headers, external sources may be grouped under the (ordered) set of target ports that they scan as they search for vulnerable hosts. In both of these examples, sources most directly correspond to IP addresses. In the context of host-layer IDS, integrity-based system monitors, and system and application logs, sources may alternatively be represented by the object access or system call invocation patterns of processes.

In general, dossier attributes can extend to features such as outcome, file access patterns (e.g., in the case of virus outbreak discovery), or commonalities in attack target configuration. Entity membership may be nonexclusive or exclusive based on tightest similarity match. For dossier-based correlation to be of value in large-scale alert repository analysis, both dossier construction complexity and volume of overall dossiers must be sublinear to the incident repository entry count, and repository update complexity must also be efficient.

In the experiments we preformed during this project, an important aspect to dossier management is the development of an applicable dossier prioritization scheme. Among these schemes, dossiers may be prioritized based on the number of common event sequences that they capture. For example a connection-pattern dossier d_1 may be considered to capture a more significant deterministic behavior pattern than dossier d_2 if the connection pattern length of d_1 is greater than that of d_2. Alternatively, one may prioritize dossiers based on attack severity, increasing the priority of d_1 over d_2 if the average severity of d_1's alerts exceed that of d_2's, regardless of their respective lengths. Additionally, membership size also plays an important role in evaluating the potential that a dossier is representing an emerging wide-spread threat versus an isolated and uncoordinated activity. For large-scale repository analysis, our interest is in evaluating the potential for dossiers to identify highly deterministic behavior patterns (such as those exhibited by automated scripts or malcode) that is emerging as a wide-spread threat.

Conceptually, it is also possible to recursively combine the dossiers produced over a set of distinct network segments into a meta-dossiers, where the entities of the meta-dossiers exhibit similar patterns of behavior to an adjustable degree of similarity. While this phase of the project did not explore meta-dossier construction, SRI is involved in a follow-on project that is exploring distributed dossier construction, with a centralized meta-dossier service.

2.2.1 N-Gram Port Analysis

One instantiation of the dossier-correlation approach is to search for deterministic behavior within the connection probing patterns of external sites as they attempt to scan one's associated IP address space. For each external source address that connects (or attempts to connect) into a monitored network, its set of target destination ports are abstracted into an N-gram port list (duplicates removed). Each unique N-gram represents a dossier, where the set of sources that match the N-gram comprise the dossier membership set.

N-Gram-based dossier correlation asserts that an emerging N-gram pattern of significant length found replicated throughout a large set of external sources indicates the presence of a deterministic algorithm or end-user probe strategy.   A longer N-gram length indicates a lower probability that the members are coincidently targeting the same set of target ports, although beyond a certain N-gram length we collapse this activity into a single dossier that represents a generic portscan. In addition, dossier membership size provides a direct indication of the breadth with which the attack probe pattern has spread among external sites.   Our experiments performed to date provide empirical evidence that the assertions hold in the analysis of blackhole data.

Connection pattern matching could also be extended beyond target port analysis. For example, a dossier could be constructed to identify deterministic patterns in the target address selection among incoming connections performed by remote sites that search the monitored address range using a small number of destination ports. A malicious application may employ a specific heuristic to select a preferential sequence in the lowest octets used to construct target addresses.

We explored dossier construction using a batch mode algorithm that assumes incoming connections are logged over a uniform temporal window. Dossiers are constructed over each temporal window, and then compared and fused to the set of dossiers constructed during previous iterations. N-gram port dossier analysis requires a minimum of data that can be collected by the egress router or data sink, or provided by a large-scale repository such as the SAN's Internet Storm Center. The minimum set of fields required for N-gram port dossier construction includes source address, destination address, source port, destination port, and protocol. Timestamps are necessary if ordered sequence analysis is performed.   In addition, dossiers can be constructed in the presence of source and destination addresses anonymization techniques, such as standard address hashing.

We performed several experiments of N-gram port dossier construction using datasets that included 115M firewall log entries and approximately 300M blackhole connection attempts.   The experiments demonstrated the ability of N-gram port dossier analysis to identify malicious bot network without a priori knowledge of these application, and provided insight into dossier prioritization methods to help distinguish highly replicated deterministic behavior from lower confidence dossiers.   We next present a summary of our experiments in N-gram port dossier construction using blackhole scan traffic.

2.2.2 Experiments on Blackhole Traffic Patterns

Blackhole traffic provides insight into general Internet scan traffic that is sent out to map an address space that is of yet unknown to the sender. Blackhole addresses refer to IP addresses that have no active device associated with them, but which may be reserved by an organization for future use. By definition, no legitimate network traffic should be sent to or from these unused addresses. Traffic sent to or from such addresses represent spoof replies or scan traffic, thus providing N-gram dossier correlation an excellent source of connection activity from which to observe automated scan tools in operation.

In this experiment, we constructed destination port N-grams per source IP addresses and identified commonalities among the N-grams produced. We employed two prioritization criteria. First, N-grams of greater length are preferred over N-gram's of shorter length, in that it is much more likely that a set of unrelated sources will coincidently scan the same 1- to 2 -gram port sequences than to probe the exact 9-gram port sequence. However, scans with very large numbers of destination ports are consolidated to represent a single port sweep dossier, as there are malcode applications that are known to scan entire destination port ranges. In addition, we observe that the number of sources that are associated with a "substantial length" N-gram provide a direct indication of the degree to which the pattern is wide-spread, and therefore representative of coordinated, or algorithmic, behavior.

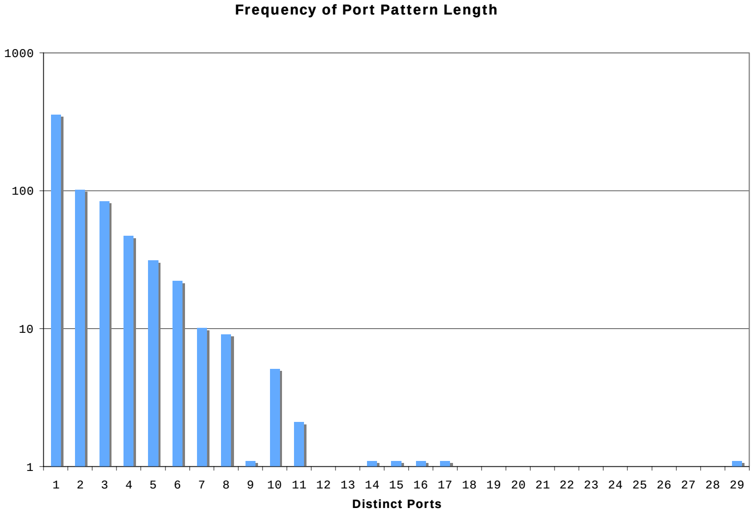


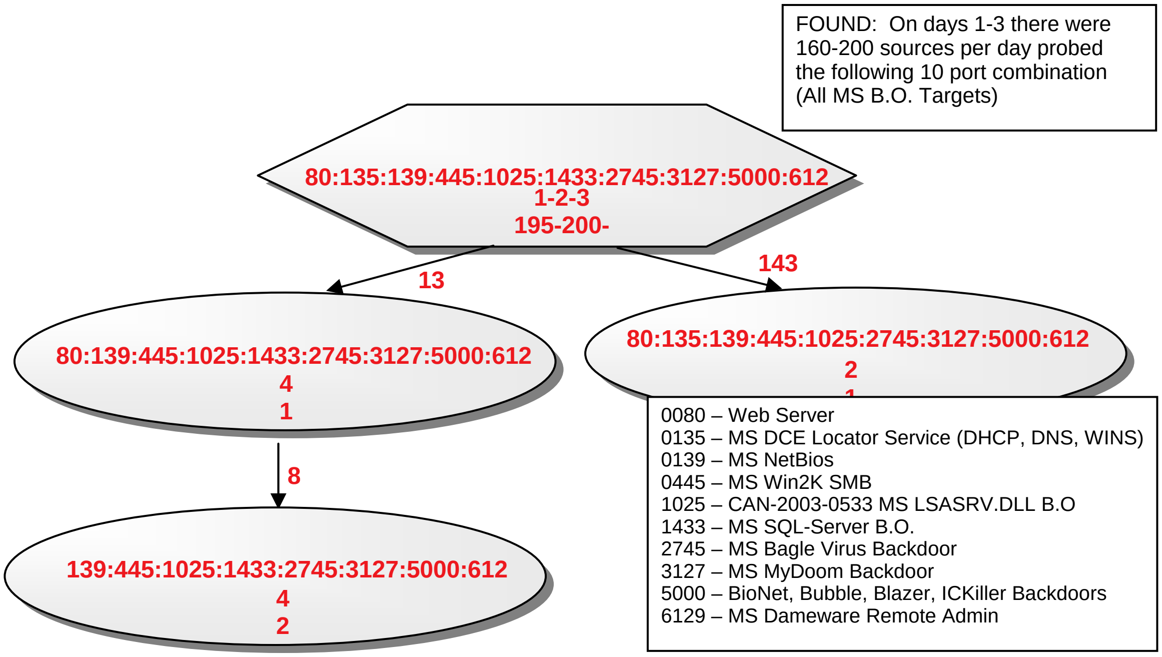
Figure 1: Sample Destination Port N-Gram Cound

Figure 1 illustrates a short selection of blackhole traffic produced by 675 remote hosts. At the extremes, 357 hosts performed a single port connection, and 1 host that performed 29 port connections (e.g., a short port sweep). In the region of 2-12 N-grams there were

313 remote hosts, and most notably there were 21 hosts between 8-11 N-grams that were found to perform slight permutations of the same port sequence (with port dropouts).

The full data analysis performed during the blackhole data experiment used a blackhole address block of roughly 56K unused IP addresses in the SRI address range of 130.107.*.* over a 36 day period. There were are total of 284M incoming scans and spoof reply packets that entered this blackhole address range during the observed time period.

One challenge in examining incoming connection patterns is the potential combinatorial explosion that exists in managing N-gram construction over a large volume of incoming connections that span a large set of source addresses and target ports. To overcome the combinatorial explosions, we downselected the number of packets to exclude packets from sources that target less than 3 unique destination ports within our blackhole address space and sources that exceed more than 100 unique destination port targets (i.e., large port scanners). The remaining packets represented a two order of magnitude reduction from the original 284M packets.



FOUND: On days 1-3 there were 160-200 sources per day probed the following 10 port combination (All MS B.O. Targets)

0080 – Web Server
0135 – MS DCE Locator Service (DHCP, DNS, WINS)
0139 – MS NetBios
0445 – MS Win2K SMB
1025 – CAN-2003-0533 MS LSASRV.DLL B.O
1433 – MS SQL-Server B.O.
2745 – MS Bagle Virus Backdoor
3127 – MS MyDoom Backdoor
5000 – BioNet, Bubble, Blazer, ICKiller Backdoors
6129 – MS Dameware Remote Admin

Figure 2: N-Gram Port Combinatorial Graph

Figure 2 provides an example of an automated port combinatorial graph constructed using our dossier correlator tool suite. The port N-gram represented in the root node represents an 11 port combination performed by several hundred remote source IP addresses over a 3-day observation window. The child nodes extending down from the root of the graph represent sources in which one or more of the original root ports were dropped. This 11-gram combination was repeated throughout the observed period of monitored blackhole traffic.

Upon closer inspection of the ports represented in Figure 2, we observed that all of the probes were targeting TCP ports that are associated with Microsoft services with known. buffer overflow vulnerabilities. It was further discovered that this 11-port combination represents the scan target list for the Phatbot malicious code backdoor scanner, and its variant Argobot. We found this malicious scanning behavior was performed consistently across hundreds of hosts during the observed time period, and were able to link these remote hosts as being infected by the same malicious code with no a priori knowledge of the bot scanning software.

2.2.3 N-Gram IDS Alert Production Analysis

Dossier correlation provides a shift in perspective in the way large-scale alert repositories are used to identify wide-spread attack phenomena. Currently, large-scale alert correlation systems focus primarily on intensity based measures, or on trends and other statistical summarizations of repository segments. Examples of these systems include SAN's DShield and Symantec's Deepsight, both of whom use their large-scale alert repositories to provide source address blacklist and long term trend analyses as their primary output.

In dossier-based IDS alert correlation, we seek to identify the emerging dominant behavior patterns in alert production from source addresses through the way in which these sources trigger signatures and anomaly detection logic. Not all automated threats will result in triggering multiple alerts from one ore more egress sensors. However, in the case of bot scanning software and worms, it is not uncommon for these tools to perform a range of probes and attacks against a target host to determine if and how it can be infiltrated. Not all automated threats employ multiple probe attempts, and those that do not are more applicable to the detection methods involving intensity-based measures than dossier correlation.

The intention of this experiment is to observe whether at least some sources do trigger multiple sensors alarms, and if so, are such alert production patterns of use in identifying a coherent attack objective. Here the dossier represents an alert production pattern that is produced by a number of remote source addresses across multiple networks. There is a potential for predictive analysis, in which an attacker who performs a subsequence of attack steps performed by other previous sources may allow us to assert with some probability that he will complete the remainder of the previously observed attack pattern. That is, if one understands that an alert sequence of size 5 is emerging across a wide range of remote attackers, then if a source is observed performing 4 of the same alerts in sequence, then one may assert with some confidences that the attacker is likely to produce the 5th attack in the sequence. This probability can be adjusted based on the 5-gram dossier membership size.

2.2.4 Experiments with IDS Alerts

We examined 146K EMERALD aggregated incident reports from an internal deployment of our EMERALD network appliance. The sensors were located on an egress switch span port monitoring all incoming traffic to the SRI/CSL network. This appliance includes three IDS sensors: Snort, eXpert-Net, and eBayes-TCP. The 146K aggregated reports were constructed from roughly 57M raw IDS alerts produced by these sensors between November 2003 and April 2004.

The aggregated security incidents represent activity from 83,382 remote source addresses, of which 82,671 sources were found to have triggered one signature. 711 remote sources triggered between 3 and 99 signatures, and 1 remote host (the SRI CIO's office) triggered over 220 IDS alert signatures during a regular site-wide vulnerability scans.

Dossier construction was performed in batch mode on all aggregated EMERALD incidents from sources that had triggered between 3 and 99 distinct alert signatures, regardless of the destination IP address. At N-gram length 4, there were 34 unique source addresses that triggered a set of N-gram variations, which included web and port-scan attacks. The sample from this single egress point demonstrates some consistency in the N-Gram pattern production, as do the N-grams at length 3, which in the presence of a mail server access problem caused 139 remote addresses to trigger variations of a 3-gram that reports email relay and service down problems. Table 1 below summarizes the findings of EMERALD alert N-gram analysis.

Table 1 - N-Grams of Observed Alert IDs

Membership Size	Alert N-Gram Length	Alert Signature IDs
2	6	BAD_ACCESS CGI_ATTACK FTP_PROBE PORT_SCAN IP_SWEEP UNUSUAL
13	4	HTTPDOTDOT BAD_ACCESS CGI_ATTACK UNUSUAL
11	4	HTTPDOTDOT BAD_ACCESS CGI_ATTACK PORT_SCAN
10	4	SYN_FLOOD PORT_SCAN IP_SWEEP UNUSUAL
11	3	HTTPDOTDOT BAD_ACCESS PORT_SCAN
10	3	EMAIL_RELAY TCP_SCAN BAD_METHOD
58	3	EMAIL_RELAY SVC_DOWN UNUSUAL
38	3	EMAIL_RELAY SVC_DOWN BAD_CMND
22	3	EMAIL_RELAY SVC_DOWN BAD_ARG
11	3	EMAIL_RELAY SVC_DOWN BAD_ACCESS
63	3	SVC_DOWN BAD_ACESS CGI_ATTACK
11	3	SVC_DOWN PROC_TABLE BAD_ACCESS
12	3	SYN_FLOOD PORT_SCAN IP_SWEEP
16	3	PORT_SCAN TCP_SCAN BAD_METHOD
37	3	PORT_SCAN IP_SWEEP UNUSUAL

However, the dataset is insufficient for performing a multi-enterprise comparative assessment of IDS alert N-gram production. Our research project was not able to acquire uncensored IDS alerts from egress points with comparable IDS tools.

2.3 Nonlinear Dynamical Analysis

Our objective in this topic area was to characterize the dynamics of packet flow in networks under a large perturbation (e.g., initial stages of a worm event). We focused on scale-free networks exhibiting the "small world" property. Our analysis began with simulation results, which were validated by analysis of actual data from an Autonomous System (AS).

A *scale-free network* is one whose topology is such that the distribution of the number of connections any given node has to other nodes in the network has no preferred average value. Rather, the distribution of the number of connections is a power- law containing an extremely broad range of values. This is not dissimilar to the property that fractals have; that is, self-similarity produces no preferred length scale. Many research publications over the last two years have determined that real computer networks obey relatively well this scale-free property. This property is important because the topology has a strong effect on the dynamics of information flow in the network, as does the small-world property.

The *small-world property* of a network refers to the relatively recently discovered, and somewhat revolutionary, property whereby the connectivity of the nodes in a network becomes dramatically greater when relatively few long-distance connections are added. In other words, by adding a few long-distance connections, it becomes far easier to move from one place to another in the network, measured by the number of hops. This property has also been shown to be a generic property of many real-world (including cyber) networks, because it is a result of self-organization as the network grows.

We observed that highly connected nodes do visibly respond to large packet flux very quickly. Also, nodes that are relatively close topologically respond visibly. The net response of a node is some combination of these effects.

We developed a simulation tool to be able to look in some depth at the details of the dynamics of packet flux in a scale-free network. The idea initially is to get a much better understanding of propagation of disturbances through a scale-free network.

We implemented an initial version of the simulator and explored some aspects of the (time varying) flux dynamics. The following are among the tasks it performs:

- Grows a scale-free network with particular connectivity index
- Offers four built-in topology generators widely used in the literature and is capable of reading in user-supplied networks as well
- Characterizes the topology and connectivity—for example, by the distribution of node 'sizes' (i.e., number of outgoing links)
- Calculates the shortest path lengths from different nodes in O(m) time (m=# of nodes)
- Simulates packet traffic on the network by using several parameters, such as birth rate and lifetime distribution
- Calculates indicators of the packet traffic—for example, time series of flux at given nodes and mean path length
- Outputs the time series from any node queried
- Calculates correlations between time series at queried nodes
- Offers enhanced visualization capabilities: it can create snapshots as well as animated movies of the state of the network

Our working hypothesis is that, because the network topology induces nonhomogeneous dynamics, certain subsets of the nodes should be somewhat correlated. In other words, a disturbance in mean flux at one node will induce disturbances at other particular nodes before they are felt in the network as a whole. Hence, these node subsets can be used to 'predict' the onset of a major network event before it would be noticed by global characteristics.

We established two early results that are of interest:

- The general weakness of the correlations. Small nodes (the vast majority) do not show any effect even when the largest node is being perturbed with packets. Larger nodes show an effect only if they happen to be not too far away. We may have to consider a different perturbation scheme than the single-node approach, or a larger perturbation magnitude.
- The effect of long-range random shortcuts (small world), which tends to destroy any correlations and drastically reduces distances. These extra links rendering it a small-world network complicate the packet diffusion significantly. It should be interesting to investigate whether finite queues and real addressing would change this at all.

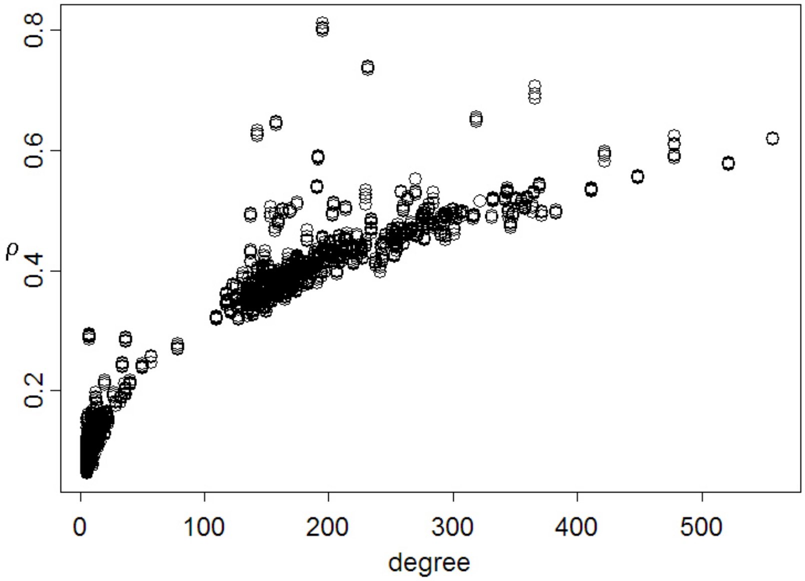


Figure 3: Correlation Versus Node Degree

Figure 3 displays the node correlations as a function of degree (number of nodes one step from a given node) in a highly clustered scale-free network with small world parameter μ=0.2.

We performed numerical experiments using the network simulator to verify recent analytic results on propagation of worm-like events in the susceptible-infected (SI) model. We found verification of a cascading effect; that is, most-connected nodes are preferentially attacked, down to least-connected nodes. This has significant implications for attack mitigation and parsimonious monitoring. Figure 4 shows (a) the time evolution of the average degree of the Newly Infected Nodes (NINs) for SI outbreaks in the AS network with 11461 nodes, and (b) and (c) cumulative fraction of infected nodes binned by degree and distance to initial seed, respectively, as a function of time.

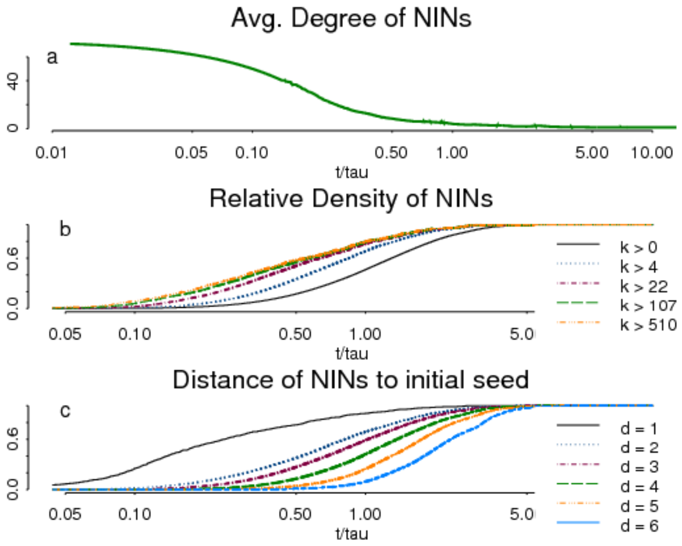


Figure 4: Infection Fraction as a Function of Node Degree, Density, and Distance to Seed

We performed numerical experiments with the NSI network simulator, using our traffic flux model to look at node correlation. This showed independent verification of the effect noted above. We also found correlation of traffic flux dynamics as a function of shortest network path length. This has significant implications for attack mitigation and network design.

Subsequent to the above, we explored what we consider a more promising approach to the enterprise-level anomaly detection problem. This entails first reducing the dimensionality of the network state space by segmentation using (possibly different) principal component-like projections of the nominal traffic dynamics. Our probabilistic estimators of hierarchical propagation are then used to determine likely infection regions. This approach will allow more rapid computation and more direct visualization/ characterization of the global system state.

We verified across a broad range of network topology and parameter types that spreading phenomena are a combination of long-time distance-driven propagation combined with short-time node hierarchy-driven selection. This will allow us eventually to generate a "risk map" for a network so that an analyst can assess potential attack scenarios.

So that we can observe the propagating phenomena in real time, we developed spatio-temporal visualization tools that we will eventually develop into more sophisticated analysis tools.

We obtained two real-world data sets with which to verify the results of our simulation. This led to further refinement of models, validating the simulation insights with data from an actual Internet Autonomous System (AS). The Internet passive measurement produces BGP AS graphs, which are constructed from Internet inter-domain Border

Gateway Protocol (BGP). The Topology Project at the University of Michigan provided the extended version of the BGP AS graph.

We have improved the predictive ability of our models by using random walk centrality in place of simple node degree. This analysis will identify points for parsimonious insertion of high-fidelity monitoring as well as enable generation of network risk maps.

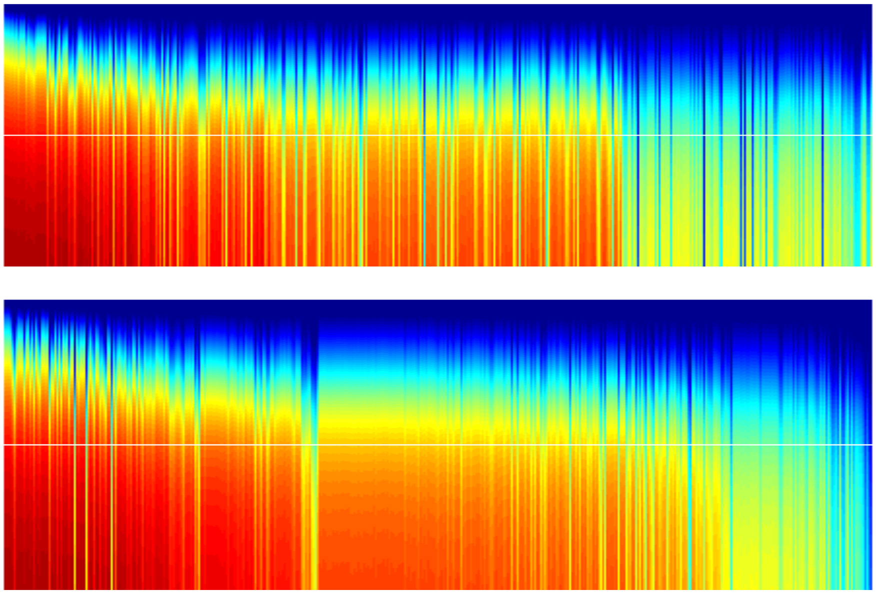


Figure 5: Evolution Infected Fraction Versus Node Degree (top) and RWC (bottom)

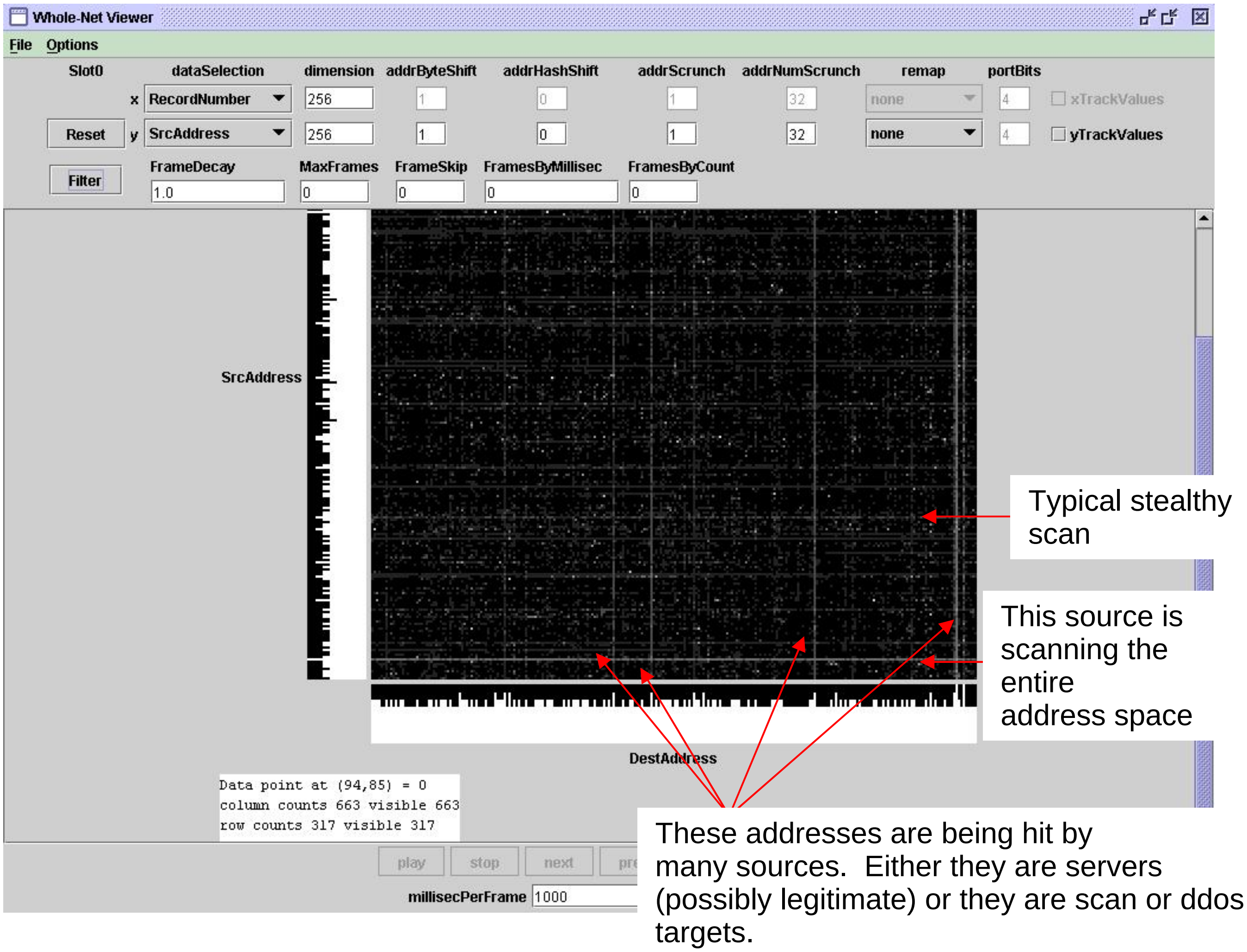
Figure 5 shows the time evolution of the fraction of infected sites as a function of descending node degree K (top) and descending Random Walk Centrality (RWC) C (bottom) of the AS network, with 11461 nodes.

This work culminated in a paper submitted to *Physical Review Letters*. The paper, “Improved Epidemic Path Predictability in Complex Networks”, was authored by Markus Loecher and Jim Kadtke, Nonlinear Solutions, Inc., March 17, 2005.

3 Technology Transition

We had several technical interchange meetings with Dr. Paul Krystosek and his staff at the Department of Energy Computer Incident Advisory Capability (DOE CIAC), at Lawrence Livermore National Laboratory. We considered that DOE has important similarities with the IC, specifically multiple independent administrative domains, but working on common missions and engaged in activity of interest to an adversary. CIAC gathers intrusion detection alerts and other cyber security relevant information for more than 100 DOE laboratories.

Unfortunately, we were not able in the end to reconcile DOE and DOD clearances, and so were not able to collaborate at a deep and detailed level. DOE CIAC did implement early prototypes of our analytical tools and ran these tools against CIAC data. Figure 6 below represents a sanitized, annotated "screen shot" of the scalable visualization tool against a day of CIAC data from November 2004.



Slot0		dataSelection	dimension	addrByteShift	addrHashShift	addrScrunch	addrNumScrunch	remap	portBits	
	x	RecordNumber	256	1	0	1	32	none	4	□ xTrackValues
	y	SrcAddress	256	1	0	1	32	none	4	□ yTrackValues

FrameDecay	MaxFrames	FrameSkip	FramesByMillisec	FramesByCount
1.0	0	0	0	0

```
Data point at (94,85) = 0
column counts 663 visible 663
row counts 317 visible 317
```

millisecPerFrame 1000

Typical stealthy scan

This source is scanning the entire address space

These addresses are being hit by many sources. Either they are servers (possibly legitimate) or they are scan or ddos targets.

Figure 6: Screenshot of Scalable Visualization, CIAC Network

We also held meetings with members of the IC at which we provided technology demonstrations. While there was some mutual interest, these efforts unfortunately did not result in an ongoing deployment.

Appendix A. Entropy Characterization of Propagating Internet Phenomena

We seek a statistical characterization of distributed and propagating phenomena that is meaningful at multiple levels (enterprise, ISP, backbone), and for which autonomous domains can exchange information with no compromise of confidentiality. We conjecture that distributed attacks and in particular propagating phenomena such as worms have different entropy from background traffic. Specifically, we investigated whether the entropy of the process of UDP destination ports and source addresses for UDP packets rejected at a firewall changes during the early stages of propagation of a worm. We observe that with the case of the January 2003 Slammer worm, the entropy of the UDP port processes declines as rejected requests become more concentrated in a single state (in this case, the target port). The entropy of the source address process increases as the client base accessing our site becomes larger and more random. Autonomous systems can exchange entropy statistics for packet streams with no confidentiality concerns, potentially enabling timely and cooperative detection of events of interest.

Although our case study is limited to our local gateway, we believe that the technique may be especially useful for consideration at the ISP or backbone level. We introduce an efficient iterative computation for entropy that may be feasible for the packet rates in such environments. The process state space is regularly aged and pruned so that the entropy estimate responds quickly to the underlying process. Pruning also maintains a manageable state space and contributes to computational efficiency.

Introduction

There have been several recent and highly publicized examples of Internet attacks consisting of self-propagating phenomena ([CE01, Mo03] describe just two of the better-known examples). The attacks consist of attempts to exploit a specific vulnerability, and when the attempt is successful the attack propagates from the infected host to other vulnerable targets. The attacks have used a variety of means to generate new addresses that are subsequently probed for the exploit. Such attacks have the potential to saturate the vulnerable population of hosts on the Internet in a very short time [Sta01, Sta02a], and the probes launched with the objective of self-propagation may significantly degrade local and Internet-wide quality of service. For example, although our site has no hosts vulnerable to the Slammer worm [Mo03] and the firewall rejects the malicious UDP packets, our users experienced significant performance degradation during the period of maximum Slammer attack intensity.

These attacks stress enterprise-level intrusion detection systems. We observed more than 100,000 alerts at our laboratory gateway router related to the Blaster and Nachi attacks [MS03], a rate of almost 1.5 alerts per second. This is in spite of alert threading by an alert correlation utility [Va01]. In order to assess whether such attacks are local or global, and perhaps take early intervention at the ISP or peering point level, the alert volume from a large number of autonomous enterprise-level IDS is probably too voluminous to be useful. Moreover, the alert messages may contain information that an enterprise is reluctant to publicize. We seek a global characterization of such attacks that might be visible at an early stage in the attack and at various scales on the Internet from the enterprise to the backbone level.

The work of Yegneswaran et al. [Ye03] is an important attempt to address the issue of global characterization of attacks of this nature. The statistics they tabulate are mostly descriptive in nature and produced on batch archives of syslog data. Like these authors, we seek a statistical characterization of these attacks that is likely to hold globally, but computable in near real time and summarized as one or a few derived metrics. Moreover, our metric, which is based on the entropy of certain fields in the packet stream, contains no confidential information and can thus be shared among the defenders without reservation.

Burnett [Bu03] proposes displaying resource usage counters with the Multi Router Traffic Grapher (MRTG), noting that many attacks manifest as changes in graphic traces of these counters, which the user identifies visually. The distribution of a certain counter, such as the return code from a Web server, may be suitable to the methods presented here,

Entropy [Sh48] measures the degree to which a process fills its available state space. In our case, this corresponds to the degree that observations are concentrated in a few categories (this is analogous to a more highly ordered state in a physical system) or more randomly dispersed over possible values. Sudden changes in entropy may be an indication of some change in the underlying process. For example, seeing a sharp increase in entropy of source addresses to a service with a fairly regular client base may indicate a new random distributed attack. Conversely, an increase in activity to a single port (which will result in a decrease in port entropy) may give an early indication of a propagating phenomenon. Entropy and changes in entropy are therefore potentially interesting from the network security standpoint, at the enterprise level or higher within the Internet as a whole.

There are, however, computational issues with computing entropy on features observed in high-speed network traffic. For one, the calculation requires calls to the log function, which is a comparatively expensive operation. In addition, the state space (e.g., distinct IP addresses) may be very large, leading to issues of tractability and state space explosion. This explosion may be mitigated to some degree by consolidating the raw categories (e.g., consider the apparent national origin of an IP address rather than the raw address itself), which may be desirable for reasons other than computational efficiency.

Staniford et al. [Sta02b] use an entropy-like measure to detect stealthy portscans in the SPICE system, going after a different class of network attack than what we pursue. The probabilities over which SPICE computes entropy are specific to hosts and services available at a domain, as opposed to the empirical distribution of observables of interest in the stream itself, as we derive here. Moreover, it is not clear that SPICE exploits iterative entropy computation and state space management comparable to what we present. Here, we outline a procedure to compute entropy iteratively with a minimal number of log calls and manage state space growth through periodic pruning. Entering this study we conjectured that the entropy of processes derived from packet traffic might change markedly in the early stages of a propagating phenomenon. To this end, we examined data for rejected UDP packets at our laboratory gateway during the period of the SQL Slammer worm [Mo03]. Our firewall permits externally originating UDP packets only on a very limited set of ports, and our LAN has no hosts vulnerable to this attack. Nonetheless, our firewall saw a large number of attempts at propagation from many infected hosts, an experience that was typical of the Internet as a whole during the peak of the attack.

We investigated whether the entropy for the source IP process would increase as many new source addresses sent malicious packets, while the port process entropy would decrease as the preponderance of UDP traffic became concentrated in the specific UDP port targeted by Slammer.

After some adjustments and noise removal (namely, removing failed accesses to port 137, as explained below) our conjecture appears justified, especially considering the port process. Process entropy responded reasonably even in the presence of bursty UDP background traffic, where a number of accesses to the same port are repeated. Employing this technique allows a security administrator to observe an iterative entropy trace rather than track the relative activity to a large number of ports.

In the following paragraphs, we review the definition of Shannon entropy and introduce our iterative formulation, which requires at most two log calls per observation. Next we review our experience considering the case of the Slammer worm. We conclude with a summary and discussion of implications of applying this technique at the ISP or backbone levels.

Background

The **Shannon entropy** of a distribution that tracks observation counts in various categories is given by

$$H = -\sum_{i=1}^{N_{CAT}} p_i \log\left(p_i\right)$$

$H =$ Shannon entropy

$p_i =$ probability of category i

$N_{CAT} =$ Number of categories

The log may be taken to any convenient base (a change of base changes *H* by a multiplicative constant). The common choice of base 2 leads to an interpretation of entropy as the expected number of bits required to express the information content of the underlying random variable.

If the process under consideration is highly ordered, the observations are concentrated in few categories. In the limit, if all observations are in the same category, the entropy is 0.

Conversely, if observations fall into categories at random, that is, according to a uniform distribution, entropy is maximal. In this case, all category probabilities are equal and we obtain

$$
\begin{aligned}
H &= -\sum_{i=1}^{N_{CAT}} p_i \log(p_i) \\
&= -\sum_{i=1}^{N_{CAT}} \frac{1}{N_{CAT}} \log\left(\frac{1}{N_{CAT}}\right) \\
&= \log(N_{CAT})
\end{aligned}
$$

This suggests a normalization of entropy so that it may be compared as the number of categories in the distribution changes. We define **normalized entropy** as

$$
H_N = \begin{cases} H / \log(N_{CAT}), N_{CAT} > 1 \\ 0, N_{CAT} = 1 \end{cases}
$$

The normalized entropy value is defined if more than one category is observed and measures the entropy of the observation relative to a uniform (it will approach unity as the distribution approaches uniform, corresponding to maximal filling of available states). It has the disadvantage that it masks the underlying number of categories; that is, two categories each with probability 0.5 and 1,000 categories each with probability 0.001 result in the same normalized entropy value of unity.

The normalization constant is recomputed only when the number of categories changes, either through observation of new categories or through category list pruning.

Iterative Algorithm

We derive a computationally efficient algorithm for iterative entropy calculation, and associated algorithms for pruning the state space list and aging process memory. The algorithm presented here can comfortably keep up with packet rates at our laboratory firewall to the Internet, and may be feasible at significantly higher packer rates. The exact algorithm requires two log calls per new observation rather than one log call for each category in the alphabet. We also introduce a very good Taylor series approximation that avoids even this modest number of log calls.

The iterative algorithm defined here begins by estimating the category probabilities as the count of observations in the category divided by the total observation count. We can then derive an expression for *H* based on observation counts.

$$
\begin{aligned}
H &= -\sum_{i=1}^{N_{CAT}} \frac{n_i}{N} \log\left(\frac{n_i}{N}\right) \\
&= -\frac{1}{N} \sum_{i=1}^{N_{CAT}} n_i (\log(n_i) - \log(N)) \\
&= -\frac{1}{N} \left[\sum_{i=1}^{N_{CAT}} n_i \log(n_i) - N \log(N)\right] \\
&= \log(N) - \frac{1}{N} \sum_{i=1}^{N_{CAT}} n_i \log(n_i)
\end{aligned}
$$

A new observation "touches" the above formula for only one category, so given a current estimate for H we can update the estimate for a new observation with as few as two log calls, one for log(N) and one specific to the observed category. We back out the current category's contribution to $\sum_{i=1}^{N_{CAT}} n_i \log(n_i)$, update its count and the total count as well as the two log terms that change, and then recompute H. Note that the first time a category is observed, its category count is 1, and its contribution to the sum portion above is 0. In this case, the algorithm requires updating the log(N) term only.

We now define an iterative algorithm based on the above derivation. If an observation represents a new category, it is appended to the list with a count of 1 and the number of categories is increased by 1. Although the algorithm requires at most two log calls per observation (as contrasted to one log call per category in the naïve implementation), it does require saving one additional term per category to store $T_i = n_i \log(n_i)$.

Initialization

$N_{CAT} = 0$ (number of categories)

$N = 0$ (number of observations)

$S = 0$ (will contain recursive $\sum n \log n$ total)

$H = 0$

Observation Processing

```
// Let $j$ index the current observed category.
// Increment N:
$N$=N+1;
If (j is new) {
        // Update number of categories
        $N_{CAT} = N_{CAT} + 1;$
        // $T_j$ is init to 0 for a new category
        $T_j = 0;$
        // Category count is 1
        $n_j = 1;$
}
else {
        // Back out contribution to n log n of this category
        $S = S - T_j;$
        // Update $n_j$, $T_j$ for this obs
        $n_j = n_j + 1;$
        $T_j = n_j \log(n_j)$
        // Recompute $S$ after the current observation
        $S = S + T_j;$
}
$H = \log(N) - S/N;$
```

The algorithm is easily extended to update multiple categories in the same update cycle if we wish to, for example, compute entropy every 100 observations. In this case, the counts are not adjusted by unity but for the category counts observed since the last update. Categories not observed in the update cycle require no adjustment.

Computing normalized entropy for each observation requires division by the log of the number of categories (number of symbols in the alphabet actually observed). This typically does not change for each observation, so that normalized entropy does not require an additional log call.

The above formulation is exact, or at least as accurate as the machine's log function. We note that most of the log calls require estimating log(x+1). A very good Taylor approximation is given by

$$\begin{aligned}\log(x+1) &\approx \log(x) + 1/x - 1/2x^2 \\ &= \log(x) + \frac{x-0.5}{x^2}\end{aligned}$$

Starting from a log call for x=10 and iterating the above formula out to x=500, the approximation differs from log(500) using the library log function by less than 0.002, or about 0.03% (three parts in ten thousand). On a Solaris 5.9 system, the library log function requires 99 cycles on average, while the above written in unoptimized C++ code requires 53 cycles.

Pruning and Process Memory

In addition to the event processing executed for each observation (rejected UDP packet, in our case), we have a periodic prune-and-age procedure. The purpose of this is to mitigate state space explosion by pruning very rare categories and also to downweight observations in the past while giving comparatively more emphasis to recent observations. The frequency of calls to the prune-and-age process, the drop threshold, and the aging constant used are all algorithm parameters that may be adjusted. The pruning procedure is done first so that the computations to resync iterative entropy are done on a possibly reduced number of categories. Pruning and aging tend to retain states that are more active or have been active more recently, which are likely to be the more interesting states. Pruning may not be necessary in the case of a fixed number of categories (e.g., the entropy in a byte stream needs to consider at most 256 categories). A pruned category that is subsequently observed again is simply appended as a new category with no adverse effect.

Pruning

```
$Input$: Threshold probability for pruning $p_t$
// Compute threshold count for pruning
$count_t = N * p_t;$
// Traverse list pruning categories with counts below threshold
for(all categories $i$){
if $(n_i < count_t)${
    $N = N - n_i;$
   $\mathrm{N_{CAT}} = \mathrm{N_{CAT}}$ -1;
    < release category $i$ >
}
}
```

At this point the pruned category list may be sorted for optimized list searches.

Aging

The aging procedure ages category and total counts, the intermediate terms required for the iterative entropy calculation, and resyncs the entropy with respect to the new list length and aged counts.

```
$Input$: Aging factor, $0 < age\_factor \le 1$
// Age the total observation count
$N = N * age\_factor;$
// Reset recursive S term
$S = 0;$
// Traverse list. Age category counts and pick up $T$ terms on the fly
for (all categories $i$) {
   $n_i = n_i * age\_factor;$
   $T_i = n_i \log(n_i);$
   $S = S + T_i;$
}
```

```
// Recompute H
$if\left(N_{CAT} > 1\right) H = \log(N) - S/N;$
$else\ H = 0;$
```

Results for the Slammer Worm, January 2003

In determining Slammer worm impact on live traffic at our Internet gateway, our conjecture was that such a propagating phenomenon manifests as an increase in source IP entropy and a decrease of UDP port entropy.

Our firewall drops nearly all externally originating UDP packets, so Slammer did not compromise any hosts at our site. Its impact was mainly in bandwidth consumption, which was significant.

We ran the algorithm from a cold start for 50,000 firewall rejected-packet messages with the prune-and-age procedure invoked every 500 records. The aging factor used was 0.5, and the threshold drop probability was 0.001. This data subset is just before our best estimate of the start of Slammer requests, and may be considered Internet background UDP noise.

Source IP entropy (normalized) ranges from 0.65 to 0.75, and after processing this data subset there are 49 source IP addresses in the process memory. The entropy value indicates a fair filling of available states, although there were five sources that each accounted for over 10% of the total, including one with a category probability of 0.24.

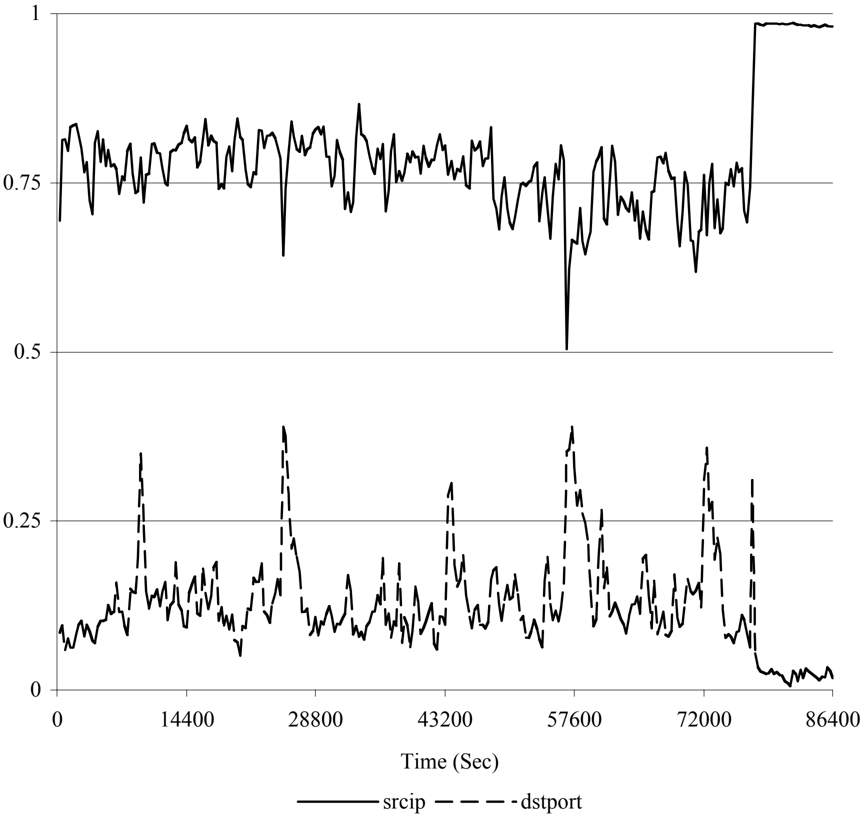
The destination port entropy was much lower, typically ranging from 0.07 to 0.15. The port category list had nine entries, with port 137 accounting for 95% of total requests and ports 53 and 38293 at a distant second and third with, respectively, 2.5% and 1.5% of the total.

It appears that port 137 requests are the dominant component of the background radiation of the Internet. The present study did not consider a cross-entropy analysis (where the joint distribution of source IP and destination port is analyzed) for reasons of state space growth. We did observe that the five most active sources in the subset all appeared to be engaged in sequential scans for port 137. The ubiquity of port 137 requests was sufficiently high that the port distribution is already quite low in entropy, so that a further lowering due to the emergence of a propagating phenomenon might be difficult to detect, or the time to detection might be delayed. This presents a challenge in that port 137 requests in some sense look like propagating phenomena with respect to a high-level measure such as process entropy. We therefore proceeded with our analysis, considering the port 137 accesses and also ignoring these requests, cognizant that we were most likely ignoring some attack traffic.

Ignoring port 137 requests for the same 50,000 record background data set resulted in slightly lower source address entropy and significantly higher destination port entropy. After pruning, only 15 sources are present in the process memory. The port list is dominated by port 53 at 46% of the total and port 38293 at 24% of the total, with normalized entropy values in the 0.4 to 0.7 range.

Figure 1 shows the normalized entropy traces for source IP and destination port process entropy for firewall log records pertaining to dropped UDP requests on January 24, 2003. The first 50,000 of these are the same as the subset discussed above for background parameter estimation. Slammer activity starts at around 77,400 (clock time 21:30). The prune-and-age process is invoked every 300 seconds with an aging factor of 0.5. Port 137 accesses are left in for the traces in Figure 1.

Figure 1: Normalized Entropy



We observe that the normalized source IP entropy increases from about 0.7 to above 0.95 (recall the theoretical limit is unity) and remains there for the duration of the trace. The pruned list has 431 source IP addresses.

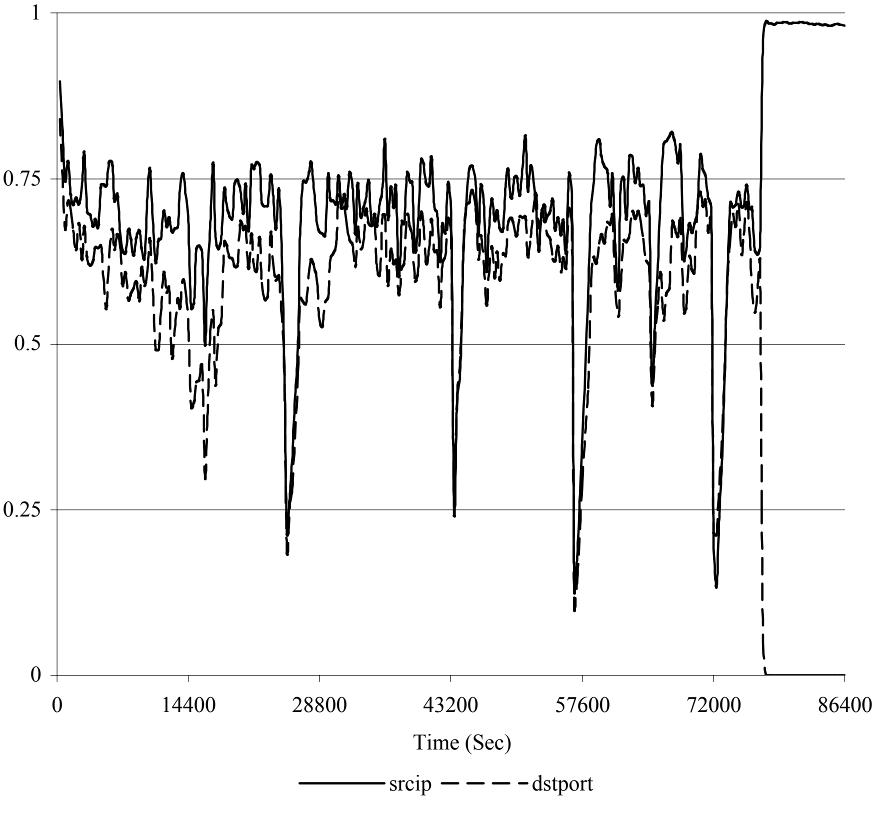
With respect to destination port, we observe that entropy first increases as the distribution ceases to be dominated by port 137, transitions to dominance by port 1434, and then remains below 0.03 for most of the rest of the trace. For some age-and-prune update cycles Slammer dominates to the point that all other port activity, including 137, is aged and pruned out and the normalized entropy reaches the lower limit of 0.

Curiously, some transient downward spikes in source address entropy are coincident with transient upward spikes in port entropy. These appear to be scans from a single source IP address, for which a burst of packets from the same source (decreasing source entropy) to ports other than 137 (increasing port entropy). We will examine this further in the future.

At the end of the trace, the only ports in the list are 1434 at over 97% and 137 at under 3%. However, if we had stopped the trace after an update cycle where port entropy was 0, and several were observed, only port 1434 would be in the list.

Figure 2 shows the same traces, but with port 137 accesses removed from consideration. The results for the normalized source IP entropy trace are qualitatively the same, and the source IP list consists of 430 addresses after all pruning. However, the trace for normalized port entropy differs significantly from Figure 1, in that before Slammer the entropy is between 0.5 and 0.75, and after Slammer it decreases very quickly to 0.03 and remains there for the duration of the trace. There is no changeover transition as observed in Figure 1. The scans observed in Figure 1 now manifest as downward spikes in both source IP and destination port entropy. The port list at the end of the trace contains port 1434 only.

Figure 2: Normalized Entropy, Port 137 Removed



It appears that, at least for Slammer, the conjectures we made in the introduction hold. Moreover, removing port 137 accesses appears to be a beneficial form of noise reduction.

Summary

It is apparent that propagating phenomena such as Slammer manifest as changes in process entropy of observable quantities in Internet traffic. Specifically, we observe that Slammer manifests as a sudden increase in source IP entropy and a dramatic decrease in destination port entropy. We introduced an efficient iterative algorithm that calculates these quantities with a minimum of calls to the expensive log function, maintains a manageable state space, and produces a metric that can be exchanged between autonomous domains with no loss of confidentiality. This may enable early detection of propagating phenomena. Moreover, the methodology identifies the states implicated in the process change (in this case, source address and destination port), permitting automated response in the form of throttling requests.

Maximum efficacy was obtained by what amounted to ad hoc noise removal in the form of discarding UDP requests for port 137. We hope to next explore a more rigorous approach to process mean removal. We observe that the iterative calculation for a new observation does not require that we add an integral count to the calculation (indeed, after aging counts are generally not integral). Instead of updating for the new observation with a count of unity, we instead update after removing the mean of the category count (which will in this case be its historical probability). The mean removal correction will be capped at a value strictly less than unity to allow the algorithm to work even when a category overwhelms the totals (as in the latter stages of the Slammer trace above).

At this point, we have defined an efficient procedure to compute a potentially useful metric. To use this as the basis for a detection algorithm, we still require tuning to determine adaptation parameters and thresholds at which an entropy change should trigger an alert. We will have to characterize what sort of nonmalicious anomalies manifest as entropy changes, in order to understand false positives from the method. We note that such anomalies may be interesting to administrators even if the underlying cause is not malicious.

We intend to explore how other attacks manifest with respect to entropy. The discovery of scans in the spikes of the entropy traces, for example, was unexpected.

We intend to examine other traffic with this technique. Candidate streams include internal-to-external packet headers and perhaps packet content (e.g., to explore the hypothesis that large sequences of null operations in malicious packets result in lower entropy).

It remains to be shown that this approach is feasible and gives a useful metric at the level of peering points, ISPs, or major electronic commerce sites. For these, it is normal for a very large number of clients to be active at any one time, but even then it is likely that a large number of client sessions will look different from the accesses triggered by malicious propagating phenomena.

References

[Bu03] Burnett, M. "MRTG for Intrusion Detection with With IIS6", www.securityfocus.com/1721, August 2003.

[CE01] CE01 CERT, "Code Red II: Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL", Incident Note IN-2001-09, Aug. 6, 2001. http://www.cert.org/incident_notes/IN-2001-09.html

[Mo03] Moore, D., Paxson, V., Savage, S., Shannon, Colleen, Staniford, S., and Weaver, N. "The Spread of the Sapphire/Slammer Worm", www.cs.berkeley.edu/~nweaver/sapphire, 2003.

[MS03]Microsoft Knowledge Base Article – 826234, "Virus Alert About the Nachi Worm", http://support.microsoft.com/default.aspx?kbid=826234, August 2003.

[Sh48] Shannon, C. E., "A Mathematical Theory of Communication", Bell Systems Technology Journal, Vol 27, 1948.

[Sta01] Staniford, S, Grim, G., Jonkman, R. "Flash Worms: Thirty Seconds to Infect the Internet", http://www.silicondefense.com/flash/

[Sta02a] Staniford, S., Paxson, V., and Weaver, N. "How to Own the Internet in Your Spare Time", Proceedings of the 11th USENIX Security Symposium, 2002.

[Sta02b] Staniford, S., Hoagland, J., McAlerney, J. "Practical Automated Detection of Stealthy Portscans", Journal of Computer Security, Vol 10, 2002.

[Va01] Valdes, A. and Skinner, K. "Probabilistic Alert Correlation", Proceedings of Recent Advances in Intrusion Detection (RAID01), Davis, CA, October 2001.

[Ye03] Yegneswaran, V., Barford, P., and Ullrich, J. "Internet Intrusions: Global Characteristics and Prevalence", SIGMETRICS03, ACM, 2003.

Appendix B. Data Cube Indexing of Large-scale Infosec Repositories

Alfonso Valdes, Martin Fong, Keith Skinner
SRI International

Abstract

Analysts examining large-scale infosec repositories for propagating network events are interested in quickly identifying temporal and spatial (IP address and/or port) regions containing interesting phenomena, or correlating events from different time periods. The size of these datasets strains current query capabilities provided by, for example, relational databases. We introduce a scalable, animated data cube representation and viewer, suitable for a broad range of observables, to permit coarse-grain detection and correlation in such data sets. We scale from the LAN to the Internet through flexible, locality-preserving hash algorithms mapping traffic source and destination (IP addresses or IP and port considered simultaneously). Data streams considered include inherently suspicious traffic such as packets rejected at a firewall, IDS alerts, or traffic to unused address space, as well as NetFlow data. We display observables as intensity plots, where X and Y coordinates are the hashed source and target address and the intensity is proportional to traffic volume. Source and target address space may or may not be the same and may or may not be mapped the same way. Propagating events have distinct visual signatures that can be enhanced through matched filtering techniques. Future work will correlate cubes efficiently through cell-by-cell multiplication. An analyst will be able to, for example, examine whether plots representing two time periods (hours or days) exhibit similar patterns. Multiplication of a cube with its transpose permits identification of nodes that respond to potentially malicious probes. These data cubes permit coarse-grained detection and correlation without expensive database queries.

Keywords
Intrusion Detection, Correlation, Visualization, propagating event detection.

Introduction

Information assurance analysts increasingly examine large data repositories representing scales from the enterprise to the level of multiple autonomous systems. These repositories can be quite large, presently on the order of hundreds of gigabytes per day (compressed) in some cases. Commercial database tools are unable to process such massive data sets. Queries with custom software to subset these files and identify events of interest require hours. Analysts examining these repositories are particularly interested in events with “broad footprint”, such as propagating events and so-called bot networks.

The analyst community needs a set of tools that quickly show what a particular data set contains, to identify address space and time slices of interest, and to correlate phenomenology across these sets. Of particular interest is a quick answer to the question of whether an event observed in near real time is similar to events in past data sets.

Clearly, such tools must scale to large address spaces and event volumes. A reasonable tradeoff involves visualization and correlation in a coarsely mapped manner, which trades off detail but allows analysis of the entire address space of interest. Of necessity this correlation and visualization is coarse grained, but experience shows that it is of sufficient fidelity to be useful.

We propose the use of a descriptive data cube as part of standard metadata for such alert repositories. The concept is to hash the address of observables to a limited set of coordinates, and represent intensity of observables on a two-dimensional display. We introduce the EMERALD WholeNet Viewer, a tool presently capable of consuming data in a variety of formats such as native firewall logs or NetFlows. The capability to analyze these data sources means that the viewer has utility at levels from enterprise to peering point.

The viewer also comprehends IDS alerts as an observable. In this case, typically the horizontal dimension is the signature (or hash of a signature) from an IDS. The capability to view IDS alerts in this matter permits visualization of alert signature patterns as they emerge.

The mapping preserves a notion of address proximity, so that sources that are close in IP address space hash to close coordinates in our display. We typically represent source on the vertical axis, and target (or other dimension of interest) on the horizontal axis.

By considering time as the third dimension, we are able to provide a data cube summary of an arbitrarily large data set. In principle it is possible to present the entire Internet in the viewer. A typical representation we have used maps address space to 256 coordinates, and takes a time slice every thirty minutes. A day of such data is thus represented as a data cube of dimension 256 x 256 x 48, with each intensity represented as a byte (256 level gray scale). The total cube size is a comparatively modest 3 MB. The cube may be assembled in real time or in batch mode, the latter requiring one comparatively costly pass through the data set.

The viewer permits a wide variety of replay, step, frame rate, and pane size controls. An analyst will typically display such a cube at a nominal rate of one frame every two seconds using the viewer, where a framer aggregates data for half an hour. This permits the analyst to scan a day of data in a little over a minute and a half.

There are numerous benefits from including a data cube of this type as part of the standard metadata for large repositories. The analyst may quickly screen data sets to identify time periods or events of likely interest for further investigation. This screening uses the metadata cube only. An additional benefit is that the address mapping inherently preserves anonymity of potentially sensitive data in the repository.

For the remainder of this paper, we will use the notation $CUBE(x,y,t)$ to represent the intensity of the observable at mapped source address *x*, mapped destination address *y*, and time slice *t*.

To analyze traffic between nodes, we can use the same or a different mapping of addresses for the horizontal and vertical axes. Notionally, therefore, an analyst may map the entire Internet to the vertical access and a class C network segment to the horizontal axis, where the horizontal axis represents one node per coordinate.

Scalable Visualization

We present techniques to view propagating network phenomena as analogous to image processing, where multiscale processing techniques and high-performance hardware coprocessors are more highly developed. We arbitrarily map sources to the vertical axis and destinations to horizontal. The techniques presented here are suitable for detection at peering points and are inherently scalable and suitable to distributed computation.

The basis of our approach, building on our previous work [Va04a, Va04b], is to represent the networks under examination as images, where coordinates are obtained by a suitable mapping function of the address space. At peering points, the source and destination address space may be considered the same. At gateways between the Internet and the enterprise, we may wish to represent sources from the network and destinations in the enterprise, perhaps using a different address mapping function for each.

We generated such images for source IP address/destination IP address and source IP address/destination port. In our earlier work, the intensity of the pixels corresponds to the count of connections rejected by our firewall for the source and target (IP address or port). We have since enhanced our capability to comprehend Pix Firewall log format and NetFlow data, as well as IDS alerts.

To obtain images 256 pixels on a side, the source and destination IPv4 addresses were hashed into an unsigned byte using the following algorithm:

```
result = address_byte[0];
for (i = 1; i < num_IPv4_bytes; i++)
        {
                result = leftShift (result, 1) ^ address_byte[i];
        }
  result = result % 256;
```

Here, leftShift () performs a 1 bit left-shift. Shifting by a different number of bits enables generation of different image sizes. The properties of this hashing algorithm include byte-order sensitivity and the use of most of the input bits.

The 2-byte port value is hashed with a specialized hash function that splits the dimension in half, with the first half containing the port numbers less than 1024 (the IANA reserved port range) and the second half containing the remainder. In both cases, the port is hashed using the modulus function, which has the relevant property that values less than 128 are maintained without modification, enabling the easy identification of various well-known services (e.g., SMTP, FTP, HTTP).

Value Tracking

Associated with each hashed value we optionally track count and value for the largest contributors (e.g., most active source addresses hashing to a particular cell) in the form of a truncated histogram (truncated in the sense that only the largest contributors are retained). While in practice a very large number of values might hash into the same index, in practice the activity at a given index is dominated by one value. Coupled with the zoom capability described below, we are able to ascertain the true value for events of interest in most cases.

It must be pointed out that with value tracking enabled the actual values are available to the analyst, which may have confidentiality repercussions, particularly with peering point data. It is also the case that for peering point data we may have a large number of values with comparable counts, which may lead to histograms with a comparatively large number of entities.

While this is potentially a problem, we have not observed a serious impact with the limited peering point data we have examined. Furthermore, it is the case that the number of histograms to be maintained is proportional to a single dimension of the cube (typically 256), not the square of this dimension. For IP addresses, one approach to state space management might be to track address values only at the “/24” level.

In the section on future work, we present a pruning and aging procedure to bound the size of the histograms during value tracking.

Value tracking must be enabled to permit the zooming feature described below.

EMERALD WholeNet Viewer

The WholeNet Viewer is a Java application to display network log data in two-dimensional gray-scale intensity plots, where the user may assign various hash-function algorithms to be applied to the data in each dimension. Currently, the user may select from source address and/or port, destination address and/or port, and alert signature number for the axes. The viewer reads raw data in such formats as Pix Firewall Log messages or NetFlows, and optionally saves and reloads data cubes.

As many as six simultaneous plots (referred to as “slots” in the GUI) may be displayed. The underlying data for the image consists of the record count for each hashed cell as determined by the parameters. The intensity (darkness) of the displayed pixel at any location is proportional to this count.

The data display consists of the two-dimensional image with a y-axis histogram on the left and an x-axis histogram below it. Below that are controls to play, pause, and rewind, and to the right of the controls is a text box that is used to display information about the frame being viewed. The user has options to invert (reverse black and white) the image or either histogram, compute logarithm before gray scale binning, remove the row-wise minimum (a form of background noise removal useful for enhancing horizontal scans), and zooming.

To zoom, the analyst selects a rectangular region over the data area, which is then expanded to fill the visible data area. Left-clicking a single data point changes the text-box content to display information about the data point (when the value-tracking checkboxes are selected, the corresponding values that contributed to the datapoint are displayed).

The analyst controls how the input data records are to be used to create the two-dimensional gray-scale plots for one slot through numerous settable parameters. These parameters define the data source for each dimension and the behavior of the hashing function and data collection for that dimension. There are also settable parameters that control how long a time interval (in data units) is consolidated per displayed frame.

The movie replay controls allow the analyst to set how long to display each frame (not to be confused with the data time interval per frame previously mentioned), as well as functions to pause, step, and back up over interesting sections.

The input data may arrive in a variety of formats, or may be loaded from a previously saved data cube.

Firewall Data

We processed log files from the outward-facing PIX firewall at our laboratory for the display in Figure 1. The underlying data consists of packets directed at unused address space (black hole data), and is thus inherently suspicious. Figure 1 shows a single time step of thirty minutes (a typical value for frame replay) for data from January 23-24, 2005.

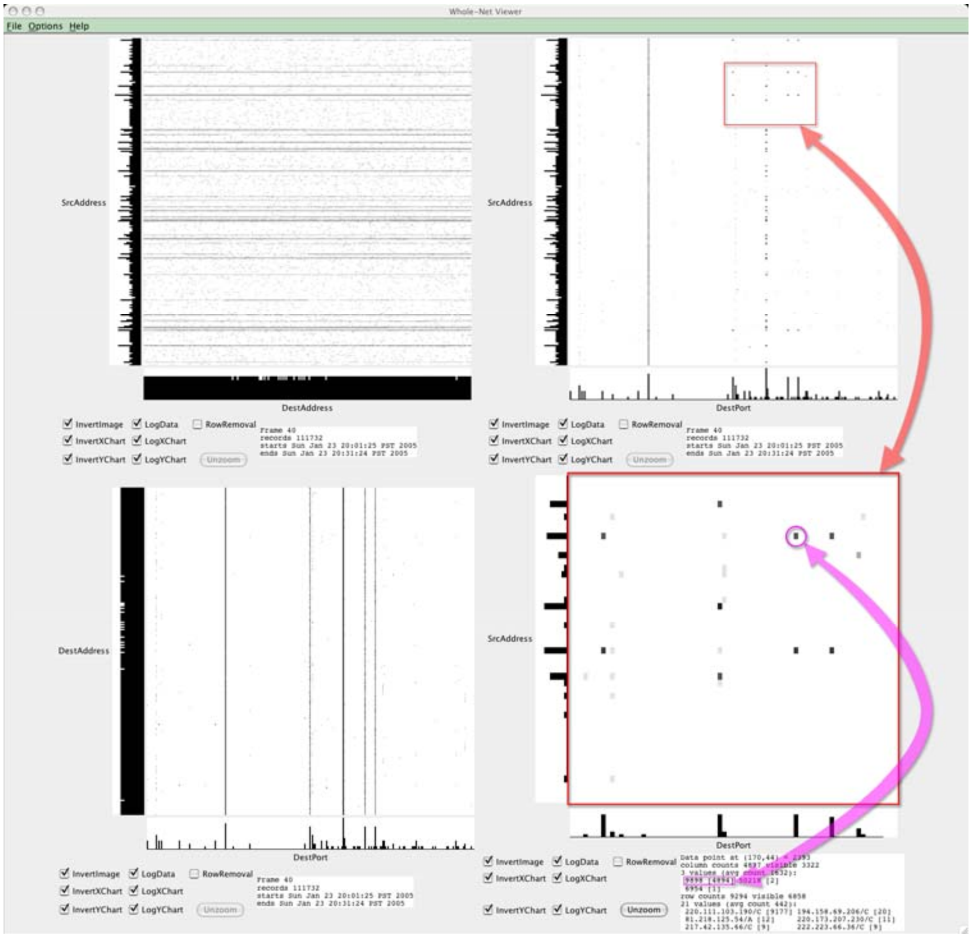
The display shows four panels, in which the rate of rejected packets is displayed in gray scale (darker values indicating a higher rate). Along the left and bottom edges of the display we also provide a histogram. For the figure shown, we observe that there is a “floor” of scanning sources but also a much smaller number of highly active sources, which show up as single bars in the left-edge histogram on the top two panels. The target address range, on the other hand, is comparatively uniform.

The top left panel shows source address to destination address. The horizontal features correspond to source addresses scanning the entire destination address space. The analyst can obtain the value of, say, ports in a bin by clicking on a pixel, usually after selecting a rectangular region and zooming.

The top right panel shows source address to destination port. Target ports are much more concentrated, with this interval showing a vertical feature (corresponding to a large number of sources) scanning for ports 445 and 1023. At 12:30 PST (frame 25), port 445 becomes dominant, and the rate of rejected packets approximately triples. At 16:00 PST (frame 32), activity on port 1433 becomes pronounced but diminishes around 19:00 PST (frame 62). However, unlike activity on port 445, which comes from a wide variety of source IP addresses, port 1433 activity is restricted to 30 or 40 IP bins (out of 256). Concurrent with this activity, at January 23, 20:00 PST (frame 40), the port triplet (1023, 9898, 5554), reflecting the Sasser.E worm, becomes noticeable in a maximum of four source IP bins. This port triplet is randomly observed until approximately January 24, 3:00 PST (frame 54).

The lower left panel shows destination address versus destination port. The features at ports 445 and 1023 are once again evident, from which the analyst would infer that these are target ports for a large number of scanners. The triplet feature of the upper right panel appears as three vertical lines in the lower left, indicating that the sources scanning for this triplet port pattern look for it over the entire target address space. The lower right panel is a zoom of the rectangular region identified in the upper right panel.

By clicking on the individual pixels, the analyst can ascertain what the actual ports are. For example, the selected hashed pixel represents 4894 scans for port 9898, two scans for port 50218, and one for port 6954 (these ports hashed into the same index, but the counts are dominated by 9898). By examining the other pixels in the triplet, we establish that these four sources are scanning in a manner consistent with Sasser.



☑ InvertImage ☑ LogData □ RowRemoval
☑ InvertXChart ☑ LogXChart
☑ InvertYChart ☑ LogYChart

```
Frame 40
records 111732
starts Sun Jan 23 20:01:25 PST 2005
ends Sun Jan 23 20:31:24 PST 2005
```

```
Data point at (170,44) = 2393
column counts 4697 visible 3322
3 values (avg count 1632):
 9899 [4894] 50218 [2]
 6954 [1]
row counts 9294 visible 6858
21 values (avg count 442):
 220.111.103.190/C [9177] 194.158.69.206/C [20]
 81.218.125.54/A [12]     220.173.207.230/C [11]
 217.42.135.66/C [9]      222.223.66.36/C [9]
```

Figure 1. EMERALD WholeNet Viewer Console

NetFlow Data

We preprocessed data from Abilene networks for one site (IPLSng) [AB05] via the flow-export module from flow-tools [FT05]. The output of this module was piped to an awk script, which extracted the timestamp, IP protocol, source and destination IP addresses and ports, number of packets, total size of packets, and duration. This result was then saved as a compressed/gzipped text file compatible with the WholeNet Viewer.

The data are all NetFlows seen at the site, not necessarily limited to suspicious flows.

We downloaded data for January 16, 2005, principally for reasons of data availability and not because of any particularly interesting phenomena of which we were aware. We did observe some apparent port scan activity.

Our java app to generate the data cube processes the Abilene data at about 34,000 records per second on a dual processor Macintosh G5 (1.8 GHz).

Alternate Views

The data represented on the horizontal axis need not be the same as that on the vertical. For example, Figure 1 shows port on the horizontal axis.

We have found it useful to examine IDS alerts in this way. These displays allow one to detect in near real time a large-scale change in the IDS alert mix. We have observed such changes in the case of large-scale propagating events. Furthermore, IDS alerts from the EMERALD Net appliance's Asset Distress Monitor can symptomatically detect events for which there is no IDS signature.

Finally, we have developed a compound map that simultaneously displays, for example, destination IP in bands of a fixed number of pixels and ports to pixels within the band, so that a given pixel in any band corresponds to the same mapped ports (Figure 2). This induces a frequency structure in the display, which is amenable to power spectrum and other frequency transform techniques. For example, attacks hitting the same pattern of ports will appear as energetic bands in the power spectrum. Frequency space techniques have long been in use in the image processing community, and highly efficient algorithms and hardware coprocessors can be brought to bear.

We can generate the displays from time slices and/or decayed moving averages. This permits near-real-time animation displays, and also ensures modest computational and memory requirements for image generation.

The displays can track the raw observable, or in a variant display the data after noise removal. Noise is estimated by continuous normalization of the underlying data (subtracting a recursive estimate of the mean and dividing by the standard deviation). If the attack process is stationary in the statistical sense, the display will appear as white noise (salt and pepper). Emerging attacks are often evident sooner in such a noise-removed display as they disrupt the underlying stationarity. This approach is effective at cuing the analyst to the novel and suspicious.

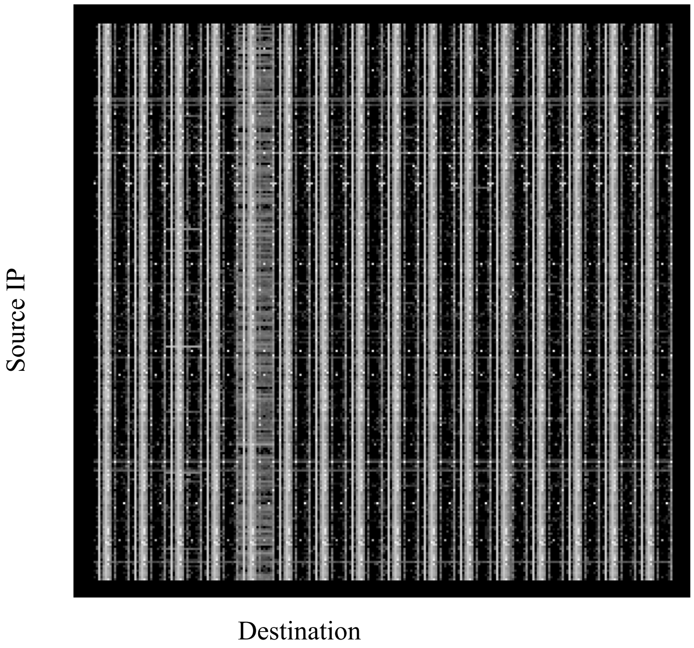


Figure 2. Source IP/Composite Destination IP/Port display, demonstrating induced periodicity

Future Work
Correlation

The data cube enables correlation across time slices, correlation from one data cube to the next, and the special case of correlation to detect bidirectional flows (e.g., an analyst may be especially interested in nodes that apparently respond to probes).

Correlation is approximated by multiplication of the intensity at a particular pair of cube coordinates. The pointwise product may then be normalized by the number of points under consideration to provide a coarse correlation estimate for two cubes.

The approximate correlation of two time slices within the same data cube is proportional to

$$CORR(t_1,t_2) \propto \sum_{x,y} CUBE(x,y,t_1) \times CUBE(x,y,t_2)$$

Computing this at various time lags provides a temporal autocorrelation function. This is defined pixel by pixel, or a rough autocorrelation can be computed by summing all pixel-wise correlations.

It is possible to correlate time slices in different data cubes, or to correlate two different data cubes. This makes sense only if both cubes use the same value map. Correlation of different data cubes (or portions thereof) is a rough measure of the similarity of phenomena in the two cubes. This permits the analyst to answer, at least in an approximate sense, whether phenomenology observed on two days is similar, or whether something similar to the present observation has been observed in the past. The correlation between cubes j and k (corresponding to, for example, days j and k in some repository) is proportional to

$$CORR(j,k) \propto \sum_{x,y,t} CUBE_j(x,y,t) \times CUBE_k(x,y,t)$$

The analyst may query the repository to identify all days that are similar (above some correlation threshold) to a cube of interest, and the computation is performed on the metadata only. The analyst may seek to identify days or time periods that correlate to a given data cube (that is, a cube containing an idealized representation of some phenomenology of interest, but does not represent any actual data).

In the case of the same mapping, the display shows which nodes connect to which. If directionality is maintained (which node initiates and responds to the TCP connection), a correlation of the figure with its transpose (achieved by simple multiplication and scaling) indicates which nodes are responding to apparent probes. Probes are ubiquitous today, and analysts have expressed a need to identify probes for which vulnerable nodes respond. Our technique presents a computationally efficient means to accomplish this.

Using a different map for source and destination, the analyst can, for example, represent the address space under his administration on the horizontal axis at much finer grain. In fact, using a display 256 by 256 pixels per time slice can represent an entire 256 node LAN segment on the target axis (horizontal in our convention).

Detrending and Detection of Inflections

A moving average estimate is computed on the fly and removed from a cube cell value. A variant is to compute moving averages with different aging constants. The difference of the short-term and long-term value per cell may indicate an inflection in the process (e.g., the early stages of a propagating event). An inflection may be more evident in a detrended cube than in the unmodified cube.

The moving average *Mov_Avg* at the initial time slice is simply the cube value. Moving averages and detrended cube values at subsequent time slices are computed as follows.

$$\text{// Input :} aging\_const \in (0,1]$$

$$Mov\_Avg(x,y,t_2) = aging\_const \times Mov\_Avg(x,y,t_1) + (1 - aging\_const) \times Cube(x,y,t_2)$$

$$Detrended\_Cube(x,y,t_2) = Cube(x,y,t_2) - Mov\_Avg(x,y,t_2)$$

Our initial analysis shows that detrending emphasizes the “loud emitters”, which is usually desirable, but that these same loud emitters sometimes overwhelm the short-term trend. When they cease their activity, detrending introduces a negative artifact. This situation appears to be analogous to signal capture in the field of signal processing. We need to look further into algorithms for correcting the trend for these emitters.

Pruning and Process Memory

We will employ a periodic prune-and-age procedure to mitigate state space explosion in value tracking by pruning very rare categories and also to downweight observations in the past while giving comparatively more emphasis to recent observations.

The frequency of calls to the prune-and-age process, the drop threshold, and the aging constant used are all algorithm parameters that may be adjusted. For large NetFlow data sets, such as from Abilene, we have a record for each flow that counts the number of packets between two addresses in a five-minute interval.

Pruning and aging tend to retain states that are more active or have been active more recently, which are likely to be the more interesting states. A pruned category that is subsequently observed again is simply appended as a new category with no adverse effect. The number of values tracked is theoretically limited to the reciprocal of the pruning probability, and is usually much less.

Pruning

```
$Input$: Threshold probability for pruning $p_t$
// Compute threshold count for pruning
$count_t = N * p_t;$
// Traverse value list pruning categories with counts below threshold
for(all values $i$){
if $(n_i < count_t)${
    $N = N - n_i;$
   $\mathrm{N_{CAT}} = \mathrm{N_{CAT}} - 1;$
    < release category $i$ >
}
}
```

At this point the pruned category list may be sorted for optimized list searches.

Aging

The aging procedure ages category and total effective (that is, aged) counts.

```
$Input$: Aging factor, $0 < age\_factor \leq 1$
// $N =$ Effective number of observations
// $n_i =$ Effective count of observations of value i
// Age the total observation count
$N = N * age\_factor;$
// Traverse list. Age value counts
for (all values $i$) {
   $n_i = n_i * age\_factor;$
}
```

From Representation to Detection

The observed patterns in actual data motivate an approach using image analysis techniques to detect phenomena of interest. Specifically, a horizontal scan appears as a horizontal feature in the source IP to destination IP view. A distributed scan to a single port appears as a vertical feature in the source IP to destination port view. If the scans come from a bot net with "close" addresses (that is, addresses that hash to near values via the algorithm previously presented), we may hypothesize that a bot scanning for a vulnerable port will appear as Figure 3 below.

A matched filter is an idealized representation of a feature that is to be searched in some image. The matched filter is convolved with the image, after both have been subjected to a fast Fourier transform (FFT). In transform space, convolution is simply pixel-by-pixel multiplication. The inverse transform then provides a matched filter field highlighting regions where the original image is similar to the matched filter.

We will explore application of matched filters to look for features of this type in the data cubes.

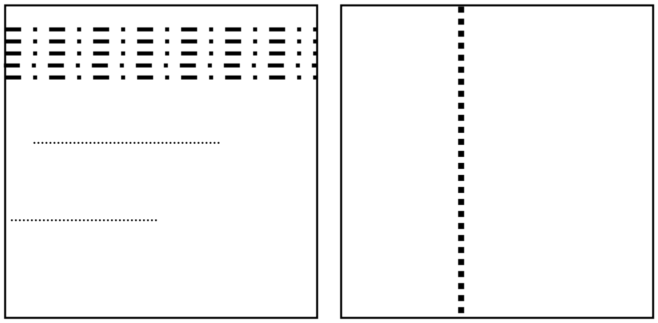


Figure 3. Hypothesized appearance of Bot Scan in Source IP/Dest IP and Source IPDest port views

Related Work

Staniford and his collaborators [Sta01, Sta02] have done important foundational work in the theoretical dynamics of propagating phenomena such as worms. Their results raise serious concerns for the Internet community in demonstrating that worm episodes can saturate the vulnerable population in an interval of time that is far shorter than the advisory/detect/patch response cycle of today.

Moore et al. [Mo03] provide a comprehensive analysis of the propagation dynamics of the Slammer worm. This attack quickly infected most of the vulnerable population on the Internet. Moreover, its propagation had a significant impact on the Internet as a whole, dominating Internet-wide UDP traffic during the period of maximum activity.

Burnett [Bu03] proposes displaying resource usage counters with the Multi Router Traffic Grapher (MRTG), noting that many attacks manifest as changes in graphic traces of these counters, which the user identifies visually. The distribution of certain count-based features such as the return code from a Web server may be suitable to the methods presented here.

Recent work in reversible sketches [Sch04] also addresses the issue of detection in large-scale networks by hashing of the address space. The use of reversible sketches presents a potential alternative to the value-tracking mechanism we have employed.

May [Ma01] detects attacks in large-scale networks by exploiting emergent properties that change statistically between normal and attack states. He employs visualization techniques for large networks, although they are different from what we consider here.

Summary

We have proposed a compact data cube as a metadata summary of very large Infosec alert repositories. We have also implemented a tool, the EMERALD WholeNet Viewer, to generate, save, and replay these cubes. We have outlined future directions whereby this capability can be extended for efficient correlation through cell-wise multiplication, and efficient detection through frequency space methods and matched filtering techniques. The viewer is implemented in Java and has been demonstrated with data formats relevant at levels from the enterprise to the peering point.

The key benefits are near-real-time visualization of interesting events, a coarse-grained but meaningful representation of extremely large data sets enabling rapid identification of regions of interest, and rapid coarse-grained correlation of interesting events.

Correlation is efficiently achieved by multiplication and scaling. We have previously described correlation of an image in which source and destination are identically mapped. In that case, potentially vulnerable destinations responding to suspicious probes are evident by correlating the image with its transpose. In general, we link an image (or a video clip of an image animation sequence) with the large files in the repository. A quick visual examination permits the analyst to identify the time slices and address ranges of interest, as desired. Also, an analyst can quickly explore if, for example, a current observation correlates with a past event.

This new approach offers an opportunity to produce an ongoing display, noise filtering, and graphical representation of observables such as incoming alerts and firewall log, giving analysts the ability to visually interpret what is happening on the Internet or in their networks. This could translate into "live" event detection, meaning faster reporting back to the victims.

References

[AB05] Abilene Networks, http://abilene.internet2.edu/

[Bu03] Burnett, Mark. “MRTG for Intrusion Detection with IIS 6”, http://www.securityfocus.com/infocus/1721

[FT05] Flow Tools, http://www.splintered.net/sw/flow-tools/

[Ma01] May, J., Peterson, J., and Bauman, J. “Attack Detection in Large Networks”. Proceedings of the Second DARPA Information Security Conference and Exposition (DISCEX II), Anaheim, CA, June 2001.

[Mo03] Moore, D., Paxson, V., Savage, S., Shannon, Colleen, Staniford, S., and Weaver, N. “The Spread of the Sapphire/Slammer Worm”, www.cs.berkeley.edu/~nweaver/sapphire, 2003.

[Sch04] Schweller, R., Gupta, A., Parsons, E., and Chen, Y. “Reversible Sketches for Efficient and Accurate Change Detection over Network Data Streams. IProc. ACM SIGCOMM IMC (2004).

[Sta01] Staniford, S, Grim, G., and Jonkman, R. “Flash Worms: Thirty Seconds to Infect the Internet”, http://www.silicondefense.com/flash/

[Sta02a] Staniford, S., Paxson, V., and Weaver, N. “How to Own the Internet in Your Spare Time”, Proc. 11th USENIX Security Symposium, 2002.

[Va04a] Valdes, A. and Fong, M. “Scalable, Signature-Free Characterizations of Propagating Internet Phenomena”, Fast abstract presented at Dependable Systems and Networks (DSN04), Florence, Italy, July 2004.

[Va04b] Valdes, A. and Fong, M. “Scalable Visualization of Propagating Internet Phenomena”, ACM CCS Workshop on Data Mining and Visualization in Computer Security, Fairfax, VA, October 2004.

Appendix C. Improved Epidemic Path Predictability in Complex Networks

Markus Loecher and Jim Kadtke
Nonlinear Solutions, Inc.

(Dated: July 8, 2005)

We apply recent results on random walkers to the analysis of idealized epidemic outbreaks in scale-free networks. By replacing the node degree with the random walk centrality we observe a refined hierarchical cascade leading to a greatly enhanced predictability for the order of infected nodes. We confirm our model results on data from real-world Internet maps at the autonomous system level. The present results are highly relevant for the advancement of dynamic and adaptive strategies that aim to mitigate network attacks.

PACS numbers: 89.75.Hc, 05.40.Fb, 05.60.Cd, 05.70.Ln, 87.19.Xx, 87.23.Ge

Studying the dynamics of the spread of diseases has a long history and is of utmost importance for the development of mitigation and containment strategies. More recently some of these results have been applied to technological networks, for example the spread of computer viruses in cyber-networks. Mathematical models usually assume a network of contacts among individuals along which the disease can be transmitted. The connectivity pattern of these networks has long been acknowledged as highly relevant to the time evolution of epidemic outbreaks as well as the final stationary state (see [1–5] and references therein).

Particularly interesting, and somewhat ubiquitous, heterogeneous connectivity patterns are found in scale-free (SF) networks. There, the number of links (degree) k attached to each node is characterized by a heavy-tailed distribution obeying a power-law of the form $P(k) \sim k^{-\gamma}$, with $2 < \gamma < 3$. As a consequence, there is no meaningful "characteristic" degree (hence the term scale-free) and more importantly the probability of finding "hubs" or "superspreaders" is not negligible. These highly connected nodes are ultimately responsible for the rapid spreading of infections and the absence of an epidemic threshold [6, 7]. This insight is of great practical interest for, e.g. computer virus diffusion and the development of optimal strategies to protect technological networks. However, the detailed dynamics of the epidemic outbreaks has received far less attention than the stationary final states of affected networks.

In this Letter, we refine and expand upon the recently observed [8] hierarchical dynamics of epidemic outbreaks in scale-free networks. While generally confirming the progressive infects hubs first and then percolates across successively smaller degree classes, we demonstrate significant improvements in the epidemic-path predictability.

We illustrate the applicability of recent results in random walk theory to obtain a more precise temporal ordering of the nodes affected by the propagating infectious cascade. In particular, the random walk centrality (RWC) [9], which utilizes global topological information to rescale the node degree, serves as a greatly enhanced predictor for the most likely cascade originating from a single randomly chosen infection seed. These findings are surprising because a spreading epidemic is best described by a many particle system while the stochastic process studied in [9] is a single random walker not exposed to any interactions. As more and more nodes become infected and turn into spreaders, the assumptions of the random walk are severely violated.

Here, we focus on the standard susceptible-infected (SI) model [1] which is widely utilized in the study of computer virus infections. Each node of the network represents an individual and each link is a connection along which the infection can spread to other systems. Individuals exist only in two discrete states, "healthy" (susceptible) or "infected". If we denote the fraction of susceptible and infected nodes at time t by *s(t)* and *i(t)*, respectively, then *s(t)* + *i(t)* = 1. At each time step, a susceptible node is infected with probability λ if it is connected to one or more infected nodes. The densities of susceptible and infected individuals can be decomposed into the respective degree classes k: $s(t)=\sum_k s_k(t)$ and $i(t)=\sum_k i_k(t)$.

It can be shown [8] that the timescale of the exponential depletion of susceptible nodes is proportional to the degree k: $s_k(t)=1-i_k(t)=s_k^0 e^{-\lambda k \Phi(t)}$. Hence, regardless of the initial density s_k^0, the fraction of infected vertices belonging to degree class k_0 will rapidly surpass all densities $i_k(t)$ of lower degree classes, $k < k_0$. The resulting hierarchical cascade was confirmed in numerical spreading experiments in BA networks [8, 10], where the initial seeds were taken to be a multitude of nodes selected at random.

We emphasize that this (global) statistical predisposition toward infecting hubs and large nodes very early in the process is clearly constrained by the local connectivities and thus by the distance to the initial seed. As the diameters of scale-free networks tend to grow at most logarithmically [11] with the number of nodes N, this "distance constraint" becomes less significant for a wide distribution of initial infection seeds. In order to fully understand the detailed spread of the epidemic, we randomly choose one node i_S as the initial seed and record both the degree *k(t)* as well as the distance *d(t)* from i_S for all Newly Infected Nodes (NINs) at time t. Here, distance between two nodes is defined as the length of the shortest path between them. We have performed numerical simulations on a large variety of scale-free networks but for brevity will show results only for a real snapshot of the Internet at the Autonomous System (AS) level as well as two recently introduced model networks [12, 13]. The AS level represents a coarse-grained description of the Internet, in which ASs are defined as independently administered domains which autonomously determine internal communications and routing policies. We are aware of the possibility of statistical bias in the AS topology due to sampling [14].

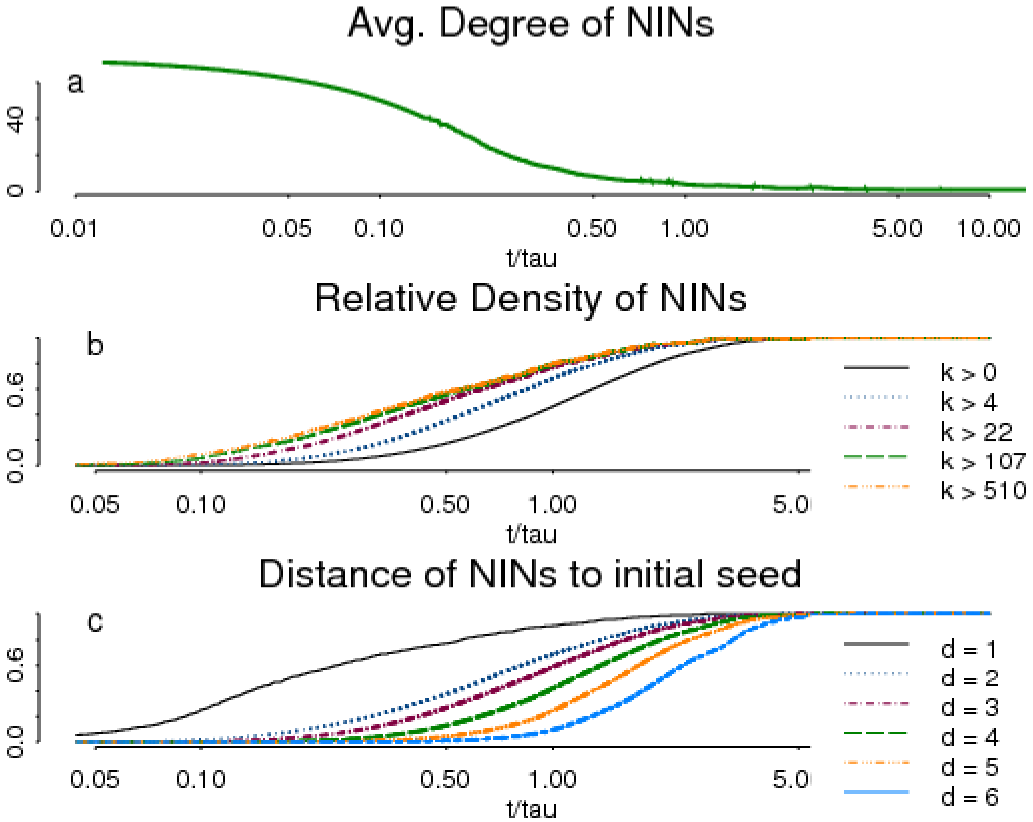


Figure 1: (color online). (a) Time behavior of the average degree of the newly infected nodes (NINs) for SI outbreaks in the AS network; $N = 11461$, $\langle k \rangle = 5.7$, $\gamma = 2.2$ and $k_{max} = 2432$. Time is rescaled by the (exponential) growth rate $\tau = 1139$ iterations for $\lambda = 0.001$. (b) and (c) Cumulative fraction of infected nodes binned by degree and distance to initial seed, respectively, as a function of time. Note that an entry in the legend in (b) such as $k > 0$ should be interpreted in the context of the remaining bins, i.e. $4 \geq k > 0$.

We summarize the results from extensive numerical simulations of SI outbreaks in the AS network in the following two figures. Figure 1(a) shows the average degree of the NINs as a function of time t, defined as [8, 10]:

$$k_{NIN}(t_n) = \frac{\sum k[i_k(t_n) - i_k(t_{n-1})]}{i(t_n) - i(t_{n-1})} \qquad (1)$$

defined only if at least one new node was infected during timestep n, i.e. if $i(t_n) > i(t_{n-1})$ [15]. The observed initial plateau corresponds to an average degree of NINs for a low density of infected vertices given by $\langle k^2 \rangle / \langle k \rangle$ [8]. After this initial quick infection of the large-degree nodes, the epidemic successively spreads along smaller nodes and $k_{NIN}(t)$

decreases smoothly with time. The lowest degree vertices tend to be reached last so that (t >> 1) converges toward the minimum degree of the network.

In Figures 1(b) and (c) we plot the time evolution of the fraction of infected nodes within classes coarse-grained by degree (b) and distance to initial seed (c), respectively. These graphs provide further insights into the interplay of distance with the statistical inclination to infect hubs very quickly. Clearly, nodes closer to the infection seed get infected earlier, irrespective of their degree. At the same time the degree distribution $P_{NN}(k)$ of the nearest neighbors of a randomly chosen node is weighted by the number of links, $P_{NN}(k) = kP(k)/\langle k \rangle$. Therefore, the average degree of the nearest neighbors of a randomly chosen node is much larger than that of the node itself which explains the observed simultaneous cascades in Figures 1(b) and (c).

Having established the existence of hierarchical propagation in a real representation of the Internet at the AS level begs the question of whether and how this added knowledge can be utilized to predict the most likely "path" along which an infection spreads. The observed hierarchical cascade implies that the degree of a node can serve as a statistical predictor of how early this node will get infected in the course of a spreading epidemic. The immediate questions to raise are (1) how good is such a prediction, and (2) are there even better approaches. Figure 2 attempts to address these points in both a visual and quantitative fashion. In the top panel we display the time evolution of $\langle i_K(t) \rangle$ for every node in the AS network by arranging the nodes in descending order of the node degree K. Note that for equal degrees we randomly assign the order of the nodes. We notice that on a macroscopic scale sorting by the node degree corresponds roughly to the temporal order of infection. However, for any fixed time, $i_K(t)$ as a function of the node degree K is nonmonotonous and somewhat irregular. We have found that a rescaled version of the node degree, which was derived in the context of random walk theory [9], serves as a much superior statistical predictor of the temporal order of infection. The computation of this predictor, which we refer to as "random walk centrality" [9] (RWC), requires knowledge of the global network topology. The RWC C of a node j is defined as the ratio of its normalized node degree $\overline{K}_j = K_j/N$ and the characteristic relaxation time $\tilde{\tau}_j = R^0_{jj}$: $C_j = \tilde{K}_j/\tilde{\tau}_j$, with $N = \sum_l K_l$ (2)

We remark that the 0th order moment matrix R^0, and thus $\tilde{\tau}_j$, can be computed via Monte Carlo simulations or analytically by utilizing the full global connectivity matrix A_{ij} of the network, defined in [16]. The RWC quantifies the potential of a node j to receive information randomly diffusing over the network.

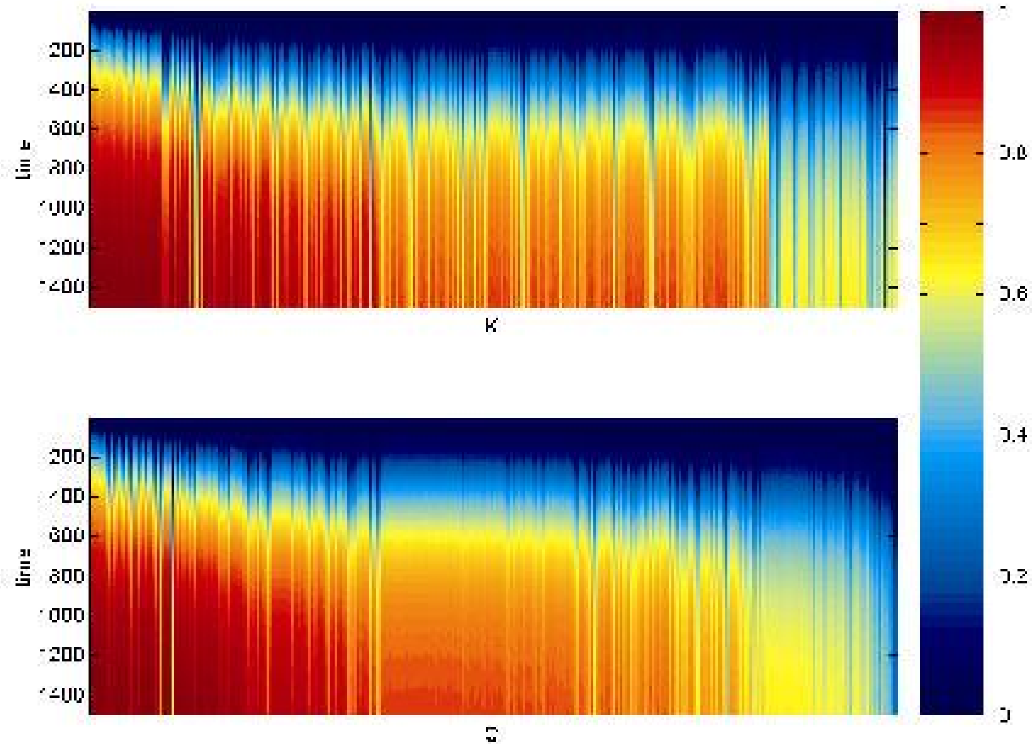


Figure 2: (color online). Time evolution of the fraction of infected sites *i(t)* as a function of descending node degree *K* (top) and descending RWC *C* (bottom) of the AS network, *N* = 11461.

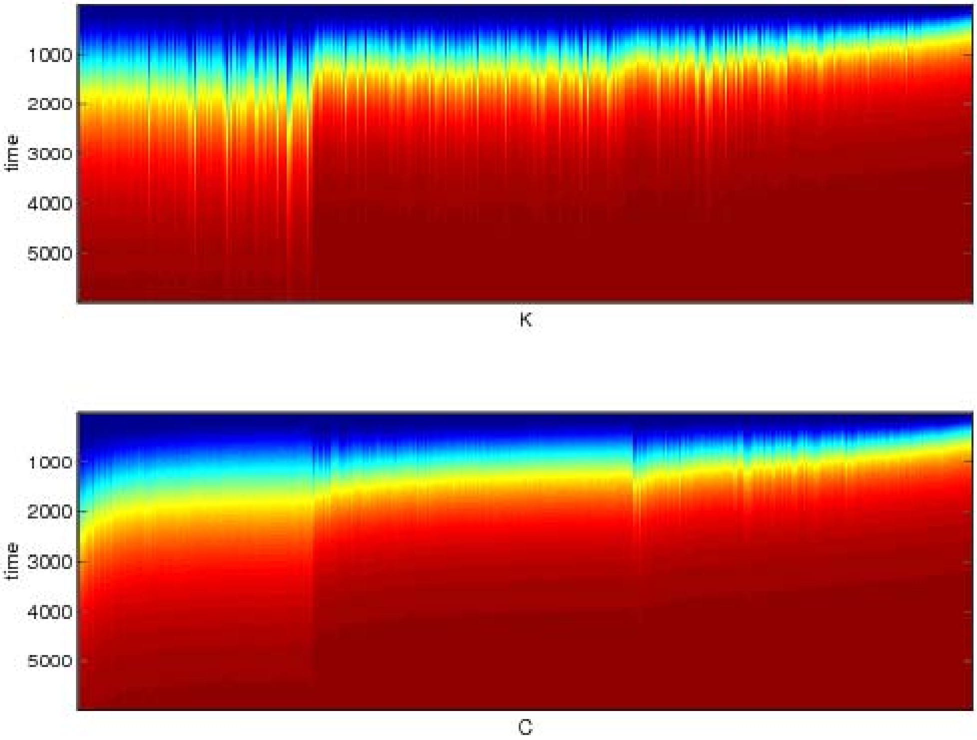


Figure 3: (color online). Time evolution of the fraction of infected sites i(t) as a function of descending node degree K (top) and descending Random Walk Centrality (RWC) C (bottom) for the PFP network with parameters $N = 1000$, $p = 0.3$, $q = 0.1$, $\delta = 0.048$. Color coding identical to Figure 2.

In the bottom panel of Figure 2 the time evolution of $\langle i_C(t) \rangle$ is displayed by arranging the nodes of the AS network in descending order of RWC. Figure 3 shows the analogous plot for a representative realization of the family of "Positive-Feedback Preference" (PFP) networks [12], which accurately reproduces most topological properties of the AS network including the rich-club connectivity. We immediately note the greatly improved smoothness of the spatiotemporal plots when ordered by the RWC. For all networks that we have investigated, $i_C(t)$ as a function of the RWC becomes much smoother and nearly monotonous. To quantify this smoothness and its impact on predictability we follow Ref. [9] and measure for each node j the time T_j at which $i_j(t)$ exceeds a certain threshold i^*. We then compute the fraction f of node pairs (l,m) satisfying $T_l < T_m$ that violate the "expected" relations $K_l > K_m$ and $C_l > C$m respectively. For the AS network, this fraction improves from $f_K = 0.16$ to $f_C = 0.04$ and for the PFP network from $f_K = 0.14$ to $f_C = 0.01$, when rearranging the nodes according to the RWC.

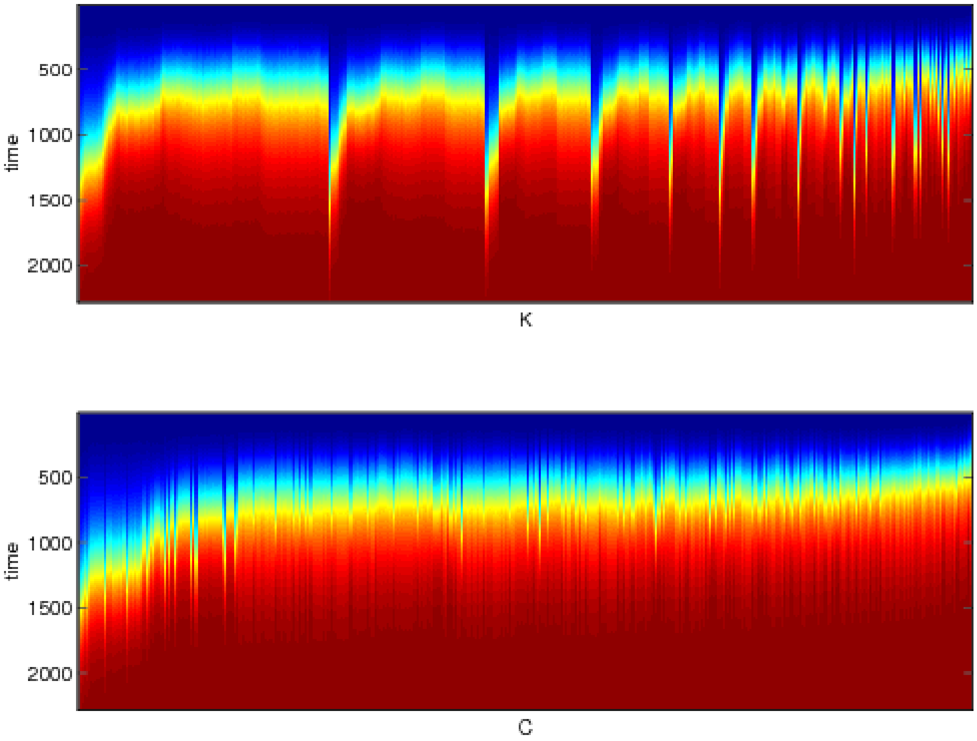


Figure 4: (color online). Time evolution of the fraction of infected sites i(t) as a function of ascending node degree *K* (top) and ascending RWC C (bottom) for the KE network with parameters *N* = 1000, m = 6. Color coding identical to Figure 2.

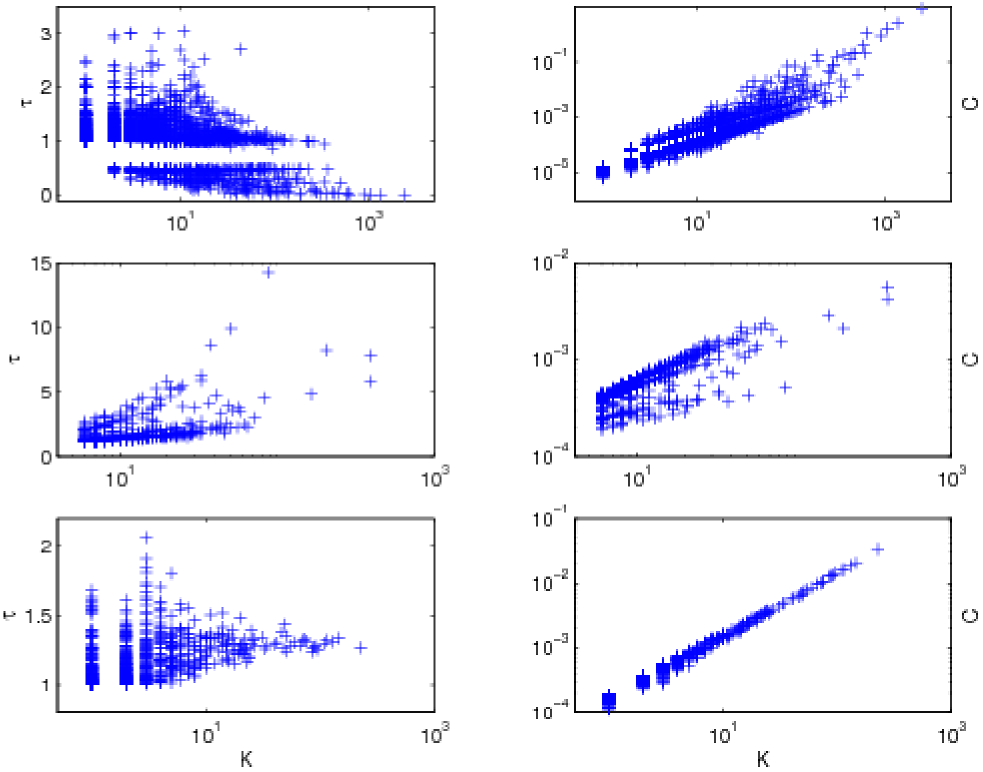


Figure 5: τ vs K and C vs K for the AS (top panel), the KE (middle panel) and the PFP network (lowest panel). Network parameters are identical to previous plots.

Recently, an interesting deactivation growth model has been proposed by Klemm and Eguyluz (KE) [13]. KE networks are highly clustered and scale free but do not exhibit the small-world property. In fact, these networks resemble regular chains with diameters that scale linearly with the network size N [17]. Although distance to the initial seed plays a much greater role in KE-networks, we nevertheless observe hierarchical propagation. We performed numerical simulations of SI outbreaks on KE networks with parameters N = 1000 and m = 6, the results of which are displayed in Figure 4. One immediately notices the bands in the upper panel that correspond to node clusters that are infected much later than their degree would suggest. The RWC adjusts for the position of these nodes, leading to a much improved temporal prediction; we find the fraction of violators to decrease from $f_K = 0.27$ to $f_C = 0.1$.

For Barabasi-Albert (BA) networks [18] with $m = 2$, τ was found to be narrowly distributed within the range $1 < τ < 2$ [9]. We observe significant differences in the relaxation time distribution among the various networks we studied. This is somewhat surprising as some models, such as the PFP growth algorithm, match all other “relevant” topological parameters of the AS network [12]. The left column of Figure 5 contains plots of τ vs K for (top to bottom) the AS, KE and PFP network. The distribution of τ for the AS network is fairly broad and contains extremely small values, τ <<1. It also displays a puzzling gap apparently excluding values in the range $0.5 < τ < 1$, which we will address in a forthcoming publication. The fluctuations in τ for KE networks also appear to be much broader than, e.g. for the BA networks, while the PFP model generates a rather narrow distribution. In the right column of Figure 5 we show the corresponding plots of *C* vs node degree. They confirm that the RWC is roughly proportional to the degree but does not increase monotonically with *K* due to the variance in τ.

In summary, we have confirmed and expanded upon the details of the hierarchical propagation of epidemic outbreaks in a snapshot of the real Internet at the AS level. We have shown that while the node degree serves as a good first approximation to predict the temporal ordering of infected nodes, the random walk centrality is much better suited for this task. While "predicting" the exact path of an epidemic is impossible due to its stochastic nature, knowing the random walk centrality can significantly reduce the uncertainties in the construction of macroscopic probabilistic maps of probable propagation paths. The usefulness of such maps can vary in scope from just outlining the most likely order of infection, to determining the most effective dynamic vaccination/containment schemes to rapidly mitigate outbreaks.

Furthermore, we have observed fundamental differences in the τ-fluctuations among networks that otherwise show great similarity in important topological features such as degree distribution, degree correlations, "betweenness" centrality, characteristic path length, etc. We therefore propose to include the relaxation time distribution, or macroscopic summaries thereof, as an additional relevant topological property of a complex network. Comparisons of models with e.g. real Internet data could therefore include the τ distributions as another macroscopic feature to match.

Acknowledgments

This work was supported by ARDA under Air Force Research Laboratory Contract No. F30602-03-C-0234. The views expressed are those of the authors and do not necessarily reflect the views of the supporting agency. The authors thank Martin Fong of Stanford Research Institute for help in calculating the large inverse matrix for the AS level analysis. M.L. wishes to thank Shi Zhou for providing the AS-Data, A. Barrat, M. Barth´elemy and A. Vespignani for valuable discussions and Jae-Dong Noh for sharing the matrix formalism used in [9].

References

[1] R. M. Anderson and R. M. May, Infectious diseases in humans (Oxford: Oxford University Press, 1992).

[2] S. Eubank, H. Guclu, A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, Nature 429, 180 (2004).

[3] H. Hethcote and J. A. Yorke, Lect. Notes Biomath. 56, 1 (1984).

[4] R. M. May and R. M. Anderson, Phil. Trans. R. Soc. Lond. B 321, 565 (1988).

[5] J. A. Yorke, H. Hethcote, and A. Nold, Sex. Transm. Dis. 5, 51 (1978).

[6] R. Pastor-Satorras and A. Vespignani, Phys. Rev. Lett. 86, 3200 (2001).

[7] R. Pastor-Satorras and A. Vespignani, Phys. Rev. E 63, 066117 (2001).

[8] M. Barth´elemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. Lett. 92, 178701 (2004).

[9] J. D. Noh and H. Rieger, Phys. Rev. Lett. 92, 118701 (2004).

[10] M. Barth´elemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, cond-mat/0410330 (2004), to appear in J. Theor. Bio

(2005).

[11] R. Cohen and S. Havlin, Phys. Rev. Lett. 90, 058701 (2003).

[12] S. Zhou and R. J. Mondragon, Phys. Rev. E 70, 066108 (2004).

[13] K. Klemm and V. M. Egu´ýluz, Phys. Rev. E 65, 036123 (2002).

[14] A. Clauset and C. Moore, Phys. Rev. Lett. 94, 018701 (2005).

[15] Averaging over a large number of temporal ensembles will reduce the times tn for which ¯kNIN(tn) is not defined.

[16] Private communications with Jae-Dong Noh; define Uji = Aij/Ki and Vji = Kj/N, then Rij = hi| [I - (U - V)]-1 |ji.

[17] A. Vazquez, M. Boguna, Y. Moreno, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. E 67, 046111 (2003).

[18] A.-L. Barab´asi and R. Albert, Science 286, 509 (1999).